

PATENT ABSTRACTS OF JAPAN

(11)Publication number : 09-319727

(43)Date of publication of application : 12.12.1997

Best Available Copy

(51)Int.CI.

G06F 15/78

G06F 11/22

G06F 11/28

G06F 13/38

(21)Application number : 08-138111

(71)Applicant : HITACHI LTD

(22)Date of filing : 31.05.1996

(72)Inventor : HASEGAWA HIRONORI

SASAKI HIROYUKI

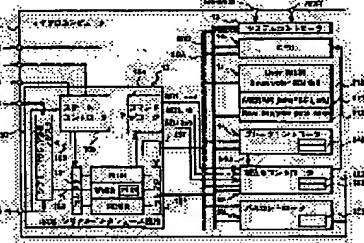
URAKUCHI MASAHIKO

(54) DATA PROCESSOR AND DATA PROCESSING SYSTEM

(57)Abstract:

PROBLEM TO BE SOLVED: To provide a data processor capable of easily varying debugging operation by reducing the number of signal terminals for debugging.

SOLUTION: The data processor 1 is provided with ROM 112 storing boot program allowing CPU 10 to transfer a debugging program from a serial interface circuit 15 to a debugging RAM area 111. When an SDI boot command is supplied from outside, the serial interface circuit outputs an SDI interruption request signal (SDI-boot) to an interruption controller 12 to allow CPU to execute the boot program. Debugging operation varies according to the contents of a down-loaded debugging program and the transfer of information at the time of debugging is executed also serially.



LEGAL STATUS

[Date of request for examination] 26.02.2003

[Date of sending the examiner's decision of rejection]

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number]

[Date of registration]

[Number of appeal against examiner's decision of rejection]

[Date of requesting appeal against examiner's decision of rejection]

[Date of extinction of right]

* NOTICES *

JPO and NCIPI are not responsible for any damages caused by the use of this translation.

1. This document has been translated by computer. So the translation may not reflect the original precisely.

2.**** shows the word which can not be translated.

3. In the drawings, any words are not translated.

CLAIMS

[Claim(s)]

[Claim 1] With a central processing unit and this central processing unit, an accessible storage means, The interrupt controller which controls interruption to said central processing unit, the serial interface means which output and input information serially between the exteriors, It is the data processor which areizing [a data processor] and changes. and the internal bus for connecting them — containing — a semiconductor integrated circuit — said storage means The 1st rewritable storage region and the 2nd storage region which holds the boot program which makes information transmit to said 1st storage region from said serial interface means to said central processing unit are included. Said serial interface means outputs the 1st interrupt request signal to said interrupt controller based on the decode result of the 1st command supplied from the outside, and said interrupt controller answers said 1st interrupt request signal. The data processor characterized by being what gives the 1st interrupt control information which performs said boot program to a central processing unit.

[Claim 2] Said 2nd storage region is a data processor according to claim 1 characterized by being what changes including the 1st vector used in order that the central processing unit which received said 1st interrupt control information may acquire the instruction address of the head of said boot program further.

[Claim 3] The information transmitted to the 1st storage region by said boot program is a data processor according to claim 1 or 2 characterized by being a program for debugging control.

[Claim 4] Said boot program is a data processor according to claim 3 characterized by being what grows into the last including the instruction which branches the instruction execution of a central processing unit to the predetermined address of said program for debugging control to which it was transmitted in said 1st storage region.

[Claim 5] It is the data processor according to claim 4 which said storage means has the 3rd rewritable storage region, and is characterized by said program for debugging control being what consists of said serial interface means including the transfer control program which makes data transmit to said 3rd storage region.

[Claim 6] It is the data processor of five claim 3 characterized by being what gives the 2nd interrupt control information which said serial interface means outputs [2nd] the 2nd interrupt request signal to said interrupt controller based on the 2nd command supplied from the outside, and said interrupt controller answers [2nd] said 2nd interrupt request signal, and makes a central processing unit execute a predetermined instruction of the program for debugging control of said 1st storage region to a central processing unit thru/or given in any 1 term.

[Claim 7] Said 1st storage region is a data processor according to claim 6 characterized by being what changes including the 2nd vector used in order that the central processing unit which received said 2nd interrupt control information may acquire the predetermined instruction address of said program for debugging control further.

[Claim 8] It is the data processor according to claim 6 or 7 characterized by being what gives the 3rd interrupt control information which said serial interface means outputs [3rd] the 3rd interrupt request signal to said interrupt controller based on the 3rd command supplied from the outside, and said

interrupt controller answers [3rd] said 3rd interrupt request signal, and makes a central processing unit execute the program of said 3rd storage region or other user rooms to a central processing unit.

[Claim 9] In the data processor which are-izing [a data processor] and changes a central processing unit and the interrupt controller which controls interruption to an accessible storage means and said central processing unit by this central processing unit — containing — a semiconductor integrated circuit — The flag made operational from the both sides of the exterior of a data processor, and said central processing unit, respectively, The data register made accessible from the exterior of a data processor, and the both sides of said central processing unit; respectively, And the command decoder which gives the interrupt request signal according to the decode result of the command supplied from the outside to said interrupt controller is included. The exterior of a data processor and the serial interface means which output and input information asynchronously and serially are established. Said storage means The 1st rewritable storage region and the 2nd storage region which holds the boot program which makes information transmit to said 1st storage region from the data register of said serial interface means to said central processing unit are included. Said interrupt controller answers the 1st interrupt request signal in said interrupt request. The central processing unit which gives the 1st interrupt control information which performs said boot program to a central processing unit, and executes said boot program The data processor characterized by being what accesses said data register on condition that said flag is in the 1st condition, and changes said flag into the 2nd condition with the completion of access to said data register.

[Claim 10] The data processor according to claim 9 characterized by being what is further equipped with the State controller which enables modification of said flag in the 1st condition with the completion of access of the data register by the data processor, and changes while said flag which it changed into the 2nd condition detects the condition of having been outputted to the exterior of a data processor and controls said data register accessible from the outside of a data processor.

[Claim 11] Data processing system characterized by being the data processor which mounts the circuit by which an access control is carried out in the circuit board, and changes by the data processor of ten claim 1 thru/or given in any 1 term, and this data processor, and being what prepares the external connection connector connected to the serial interface means included in said data processor in said circuit board, and grows into it.

[Translation done.]

* NOTICES *

JPO and NCIPI are not responsible for any damages caused by the use of this translation.

1. This document has been translated by computer. So the translation may not reflect the original precisely.

2.**** shows the word which can not be translated.

3. In the drawings, any words are not translated.

DETAILED DESCRIPTION

[Detailed Description of the Invention]

[0001]

[Field of the Invention] This invention applies a microcomputer application system to the thing of the conditions which debug under real operating environments; such as the outdoors, concerning the data processor which has debugging exchange functions, such as data processors, such as a microcomputer,

a digital signal processor, a microcontroller, and a microprocessor, especially an emulation, and relates to an effective technique.

[0002]

[Description of the Prior Art] The microcomputer for evaluation which has a debugging exchange function can be used for debugging of a microcomputer application device (it is also called the target system). In order to carry out the interface of the microcomputer for this evaluation to the host system for debugging, the signal line of a large number which may transmit data equivalent to the interface of the microcomputer and the target system concerned, the address, and a control signal in juxtaposition can be used. However, when the target system is a small system like a pocket communication terminal, it is substantially impossible to prepare the terminal or connector for connecting many signal lines for debugging to the circuit board in many cases.

[0003] Then, the technique of using serial interface for an interface with the host system for debugging is offered by "DSP Floating Point Dual-Port Processor User's Manual SECTION10 ON-CHIP EMULATOR" of IEEE issue in December, 1989.

[0004] The technique indicated by this makes the serial interface for debugging which outputs and inputs data serially synchronizing with the command decoder for debugging, a break point control circuit, and a clock signal build in, gives the command for debugging from the outside to a digital signal processor through this serial interface, decodes that command to it by the command decoder, and performs read/write of an internal register, and break point control to it according to that decode result.

[0005]

[Problem(s) to be Solved by the Invention] However, if it is in the above-mentioned conventional technique, since it opts for the function of the command for debugging fixed beforehand, the internal state which can be set as a data processor in simulation, the internal state of the data processor which can be referred to for debugging, etc. are restricted in debugging. Data processors, such as a digital signal processor, do not have more good user-friendliness in the debugging exchange function to receive such a limit, if it takes into consideration being applied to various data processing system general-purpose. If it puts in another way, there will be no variability in the actuation for debugging.

[0006] Moreover, although the connection terminal with the host equipment for debugging could be reduced by adopting said serial interface, in order to have made the number of signals into the minimum, the need of taking into consideration also about the technique of handshake control at the time of outputting and inputting information asynchronously and serially with the exterior of a data processor was found out.

[0007] The purpose of this invention is to offer the data processor which can lessen the signal terminal for debugging as much as possible.

[0008] Another purpose of this invention is to offer the data processor to which it can carry out adjustable [of the debugging actuation] easily.

[0009] Still more nearly another purpose of this invention is to offer the data processing system using the data processor which makes debugging possible in the environment near the actual use environment of the system (product system), and the data processor concerned.

[0010] The other purposes and the new description will become clear from description and the accompanying drawing of this specification along [said] this invention.

[0011]

[Means for Solving the Problem] It will be as follows if the outline of a typical thing is briefly explained among invention indicated in this application.

[0012] namely, an internal bus (16) for a data processor (1) to connect an accessible storage means (11), the interrupt controller (12) which controls interruption to said central processing unit, the serial interface means (15) which output and input information serially between the exteriors, and them with a central processing unit (10) and this central processing unit — containing — a semiconductor integrated circuit — are — izing and it changes. Said said storage means (11) The 1st rewritable storage region (111),

The '2nd storage region (112) which holds the boot program which makes information transmit to said 1st storage region from said serial interface means is included in said central processing unit. Said serial interface means (15) Based on the decode result of the 1st supplied command (SDI boot command), the 1st interrupt request signal (SDI_boot) is outputted to said interrupt controller from the outside. Said interrupt controller (12) answers said 1st interrupt request signal, and gives the 1st interrupt control information which makes a central processing unit execute said boot program.

[0013] A data processor is made to perform the user program made applicable to debugging in the application system (target system) of said data processor a software WEDE bag or when carrying out system debug, and it debugs by following the activation locus of the user program concerned, referring to the information acquired in the process externally. In order to refer to the information which appears in the activation process of such a user program externally at this time or to set up the operating condition of a user program etc., it is needed to make a data processor perform a different program for debugging control from a user program. Said boot program is a program of a central processing unit for the storing field of a user program to download such a program for debugging control to a different field of operation. By creating beforehand the program for debugging control needed according to the structure of a system for debugging to necessary, and publishing said 1st command for a serial interface means from host equipment, a data processor starts a boot program and downloads the program for debugging control supplied to a serial interface means to the 1st storage region. Therefore, when referring to the information which appears in the activation process of a user program externally on the occasion of debugging or setting up the operating condition of a user program etc., the contents of the actuation will be performed according to the contents of the downloaded program for debugging control. Thereby, any debugging actuation can be performed according to the contents of the program, and the variability of debugging actuation can be guaranteed. The debugging actuation by the data processor is not restricted by the class of command receivable [with a serial interface means].

[0014] A serial interface means is used for an interface with a host system, the number of the terminals for interfaces for debugging is lessened, and it can be compatible in the variability of debugging, and good user-friendliness under such a limit.

[0015] Also when bringing system debug or software debugging close to a real operating environment as much as possible and performing it to it outdoors to the target system like a pocket communication terminal or a pocket device, there are few terminals for interfaces for debugging, and since the variability and the good user-friendliness of debugging are also guaranteed, it is rich in mobility. [say / that it can respond immediately].

[0016] As a concrete mode of the above-mentioned means, said 2nd storage region can include the field (112A) which stores the 1st vector used in order that the central processing unit which received said 1st interrupt control information may acquire the instruction address of the head of said boot program further.

[0017] Moreover, the instruction which branches the instruction execution of a central processing unit to the predetermined address of said program for debugging control to which it was transmitted in said 1st storage region can be included in the last at said boot program. The downloaded program for debugging control can be started automatically by this, and first stage-control action can be started immediately.

[0018] If the download of a user program made applicable to debugging as first stage-actuation is assumed, said storage means has the 3rd rewritable storage region (110), and said program for debugging control can contain the transfer control program which makes a user program etc. transmit to said 3rd storage region.

[0019] since a user program can be arranged in the memory inside a data processor on the occasion of debugging — actually — ** — a user program can be performed with the same working speed for debugging.

[0020] In order to enable it to perform to arbitration actuation (breaking actuation) which switches the

program of a central processing unit of operation to said control program for debugging through a serial interface means on the way of [of a user program / activation] etc. from a host equipment side Said serial interface means is based on the 2nd command (SDI breaking command) supplied from the outside. The 2nd interrupt request signal (SDI_{brk}) is outputted to said interrupt controller. Said interrupt controller What is necessary is to answer said 2nd interrupt request signal and just to give the 2nd interrupt control information which makes a central processing unit execute a predetermined instruction of the program for debugging control of said 1st storage region to a central processing unit.

[0021] At this time, the field (111B) which stores in said 1st storage region the 2nd vector used in order that the central processing unit which received said 2nd interrupt control information may acquire the predetermined instruction address of said program for debugging control can be assigned and set. According to said program for debugging control, since this field is a field which can rewrite a central processing unit, it can also determine as arbitration the debugging processing which you want to perform by issue of such a breaking command.

[0022] In order to enable it to interrupt arbitration through a serial interface means on the way of [of a user program / activation] etc. from a host equipment side Said serial interface means is based on the 3rd command (SDI interruption command) supplied from the outside. The 3rd interrupt request signal (SDI_{int}) is outputted to said interrupt controller. Said interrupt controller Said 3rd interrupt request signal is answered and the 3rd interrupt control information which makes a central processing unit execute the program of said 3rd storage region or other user rooms can be given to a central processing unit.

[0023] In the configuration of the data processor which exchanges the information for debugging actuation with host equipment through a serial interface means, the data processor (1) by another viewpoint which paid its attention to the serial interface means is semiconductor-integrated-circuit-ized including the central processing unit (10), the interrupt controller (12) which controls interruption to an accessible storage means (11) and said central processing unit with this central processing unit, and the serial interface means (15). Information is serially outputted [said serial interface means / it is asynchronous and] from the both sides of the exterior of a data processor, and said central processing unit, and inputted with the exterior of a data processor including the flag (FLG) made operational, respectively, the data register (SDDR) made accessible from the exterior of a data processor, and the both sides of said central processing unit, respectively, and the command decoder (154) which gives the interrupt request signal according to the decode result of the command supplied from the outside to said interrupt controller. At this time, including the 1st storage region (111) which can rewrite said storage means, and the 2nd storage region (112) which holds the boot program which makes information transmit to said 1st storage region from the data register of said serial interface means to said central processing unit, said interrupt controller answers the 1st interrupt request signal (SDI_{boot}) in said interrupt request, and gives the 1st interrupt control information which makes a central processing unit execute said boot program. When exchanging data between serial interface means, the central processing unit which performs said boot program accesses said data register, on condition that said flag (FLG) is in the 1st condition, and changes said flag into the 2nd condition with the completion of access to said data register.

[0024] Thus, when outputting and inputting information asynchronously and serially with the exterior of a data processor through a serial interface means, by permitting access of the data register by the central processing unit according to the condition of a flag (FLG), the exaggerated light of the data register (SDDR) asynchronous and accessed by the exterior and the central processing unit of a data processor, can be prevented, and data can be serially exchanged asynchronously and easily between the exteriors.

[0025] What is necessary is to access a data register and just to make the 1st condition reverse a flag with the completion of access, when changing the flag into the 2nd condition in the exterior of a data processor.

[0026] In order not to use an exclusive signal line for outputting the information on a flag to the exterior

of a data processor but to use the signal line for serial data I/O While said flag which it changed into the 2nd condition detects the condition of having been outputted to the exterior of a data processor and controls said data register accessible from the outside of a data processor then What is necessary is just to adopt the State controller (153) which enables modification of said flag in the 1st condition with the completion of access of the data register by the data processor.

[0027] What is necessary is just to prepare the external connection connector (51) for connecting to a debug tool etc. the serial interface means included in said data processor in the circuit board (50) of data processing system, in order to enable connection of said serial interface means with the debug tool or host system for debugging control to the data processing system constituted with the application of the above-mentioned data processor.

[0028]

[Embodiment of the Invention]

Outline>> of <<microcomputer The block diagram of the microcomputer 1 concerning an example of this invention is shown in drawing 1. Although especially the microcomputer 1 shown in this drawing is not restricted, it is formed in one semi-conductor substrate like single crystal silicon of the well-known semiconductor integrated circuit manufacturing technology.

[0029] A microcomputer 1 is equipped with the accessible internal memory 11, the interrupt controller 12 which controls interruption to said CPU10, a bus controller 13, the breaking controller 14, a system controller 17, and the serial interface circuit (SDI) 15 that output and input information serially between the exteriors by the central processing unit (it is also called CPU) 10 and this CPU, and they are connected by internal data bus 16D and internal address bus 16A shown typically. It is also possible to constitute the microcomputers 1 including the sum-of-products computing element for digital signal processing etc. in the circuit block shown as said CPU10 as a data processor like the digital signal processor which specialized at digital signal processing.

[0030] Although especially the microcomputer 1 shown in drawing 1 is not restricted, it is a microcomputer equipped with the debugging exchange function. The above, the breaking controller 14, and the serial interface circuit 15 are specially carried, in order to support debugging. The microcomputer which is not equipped with the debugging exchange function is also called a real chip for convenience. The microcomputer equipped with the debugging exchange function on the assumption that a function equivalent to a real chip is also called EBACHIPPU (evaluation chip) for convenience.

[0031] The internal memory 11 of the microcomputer 1 as EBACHIPPU offers the storage region for storing the program of said CPU10 which a real chip offers by ROM (read only memory) in many cases of operation by RAM (random access memory). The field of this RAM is illustrated as a user memory area (User_MEM) 110. The field shown by 111 in an internal memory 11 is assigned to storing of the program for control for debugging (debug program), or other control information, and let it be the RAM field (ASERAM) for debugging which consists of RAM. The field shown by the reference mark 112 is a boot program field (Boot_Program) which holds the boot program for making the control information of a debug program or others transmit to said RAM field 111 for debugging from said serial interface circuit 15 to said CPU10 etc. This boot program field 112 consists of ROMs.

[0032] Said bus controller 13 performs data-access width of face, insertion control of a wait cycle, etc. according to an access object domain on the occasion of the bus access of CPU10 to the inside and outside of a microcomputer. The control mode for every address range by this bus controller 13 is determined by the value which CPU10 set as the register 130.

[0033] Said breaking controller 14 has the register 140 with which CPU10 sets up breaking conditions in the breaking mode mentioned later, it supervises whether the condition are in agreement with breaking conditions set as this register 140, such as the program address and a data value, appears in internal data bus 16D, internal address bus 16A, etc., and when the condition are in agreement with breaking conditions detects, interrupts and gives a breaking interrupt-request signal brks to a controller 12 in the user mode which mentions later.

[0034] Said interruption controller 12 performs interrupt control to CPU10. While interrupt request signal SDI_boot, SDI_int, and SDI_brk which are outputted from said breaking interrupt request signal brks and said serial interface circuit 15 are supplied, the external interrupt request signal which omits illustration, and the internal interrupt request signal which similarly omits illustration are supplied to the interruption controller 12. With reference to the interrupt priority corresponding to the demand, if the interruption controller 12 has the logical circuit 121 which determines the priority of each interrupt request and has an interrupt request while it is equipped with the logical circuit 120 to which interrupt with an interrupt request and factor information is made to correspond, an interrupt request will be received and the interruption factor information and the interrupt signal which answer the received interrupt request will be given to CPU10 so that priority may be given to high interruption processing of a priority. CPU10 calculates the vector address from the interruption factor information, and branches the processing to the instruction address specified by the interrupt vector table which acquired and acquired the interrupt vector table by it. At this time, it cannot be overemphasized that information which needs CPU10 for a return from interruption processing is evacuated beforehand.

[0035] <<serial interface circuit>> Said serial interface circuits 15 are the exterior of a microcomputer 1, and a circuit which are asynchronous, and output and input information serially, and let them be the circuits for debugging for carrying out the interface of the microcomputer 1 to a host system (debug tool) 4 through the serial controller 3 on the occasion of debugging using a microcomputer 1 of a system (target system) 5 so that it may be illustrated by drawing 2. What is shown by 2 in drawing 2 is external memory mounted in the target system 5, and this external memory 2 is connected to the bus controller 13 through the input/output port 19 which omitted illustration by drawing 1.

[0036] As shown in drawing 1, said serial interface circuit 15 has the shift parallel conversion register 150, a selector 151, a command register SDIR, a status register SDSR, a data register SDDR, a selector 152, the State controller 153 that controls the data I/O by the side of said serial controller 3, the register controller 157 which controls the data I/O by the side of CPU10, and a command decoder 154. Although not restricted especially, 32 bits, a command register SDIR, and a status register SDSR are made into 16 bits for the shift parallel conversion register 150 and a data register SDDR.

[0037] Said shift parallel conversion register 150 It considers as the shift register equipped with the function of a serial input-parallel output (serial-in/parallel-out) and a parallel input-serial output (parallel-in/serial-out). Synchronizing with the shift clock 155, the sequential shift of the serial data supplied from the serial data input terminal TDI is carried out. A sequential shift is carried out synchronizing with said shift clock 155, and the serial output of the parallel data which make an output possible into a bus 156 as parallel data, and are supplied from a bus 156 is made possible from the serial data output terminal TDO.

[0038] Said selector 151 connects alternatively one data input/output terminal of a command register SDIR, a status register SDSR, and a data register SDDR to said bus 156. Said selector 152 connects alternatively the data input/output terminal of another side of these registers SDIR, SDSR, and SDDR to said data bus 16D.

[0039] Said register controller 157 receives address information, a read/write signal, etc. which are supplied through a bus controller 13, decodes the address signal, generates the selection signal of said registers SDIR, SDDR, and SDSR, and the read/write signal over a register, and controls said selector 152. Thereby, CPU10 can access said registers SDIR, SDSR, and SDDR if needed.

[0040] A boot command, a SDI interruption command, or a SDI breaking command is supplied to the triplet of TS0, TS1, and TS2 so that said command register SDIR may be illustrated by drawing 3. if said command decoder 154 will interrupt if a boot command is supplied, and it gives interrupt request signal (SDI boot interrupt request signal) SDI_boot to a controller 12 and a SDI interruption command is supplied similarly — interrupt request signal (SDI interrupt request signal) SDI_int — moreover, supply of a SDI breaking command gives interrupt request signal (SDI breaking interrupt request signal) SDI_brk to the interruption controller 12. The contents of the processing which answers these interrupt requests

are explained in full detail later.

[0041] Access to the serial interface circuit 15 which synchronized with the transfer clock TCK by said serial controller 3, and access to the serial interface circuit 15 by CPU10 are mutually carried out with asynchronous. In order to control delivery of data mutually at this time, said status register SDSR has the 1-bit flag FLG. Let this flag FLG be the reference information for making controllable exclusively access of CPU10 to a data register SDDR, and access from the serial controller 3 side. That is, this flag FLG shows that access of the data register SDDR by CPU10 is possible with that logical value 0, and shows with it that access of the data register SDDR by the serial controller 3 is impossible. Moreover, Flag FLG shows that access of the data register SDDR by CPU10 is impossible with the logical value 1, and shows with it that access of the data register SDDR by said serial controller 3 is possible. In addition, the status register SDSR is equipped also with the bit of others which show the internal state of the SHIRUARU interface circuitry 15.

[0042] The serial controller 3 controls the data transfer to the serial interface circuit 15 by observing said flag FLG. If it detects that Flag FLG is made into a logical value 1 through the serial data output terminal TDO, the serial controller 3 will perform download which supplies data to the serial data input terminal TDI, and upload which outputs data from the serial data output terminal TDO. The serial controller 3 does not perform data transfer, unless Flag FLG is made into a logical value 1. That is, access to a data register SDDR is not performed. If said flag FLG is made into a logical value 1, access of it to a data register SDDR will be made impossible, for example, as for CPU10, the wait of the access to a data register SDDR is carried out by the bus controller 13. The serial controller 3 will clear said flag FLG to a logical value 0, if data transfer is completed. If Flag FLG is cleared by the logical value 0, access of the data register SDDR by CPU10 by which the wait was carried out will be resumed, and a lead or light access of as opposed to a data register SDDR in CPU10 will be enabled. CPU10 sets Flag FLG to a logical value 1 after the completion of access of the data register SDDR by CPU10. When the serial controller 3 detects this condition, the serial controller 3 is again made accessible in the data register SDDR of the serial interface circuit 15.

[0043] In short, the serial controller 3 accesses a data register SDDR, when Flag FLG is a logical value 1, and it clears Flag FLG to a logical value 0 by the completion of access. CPU10 accesses a data register SDDR, when Flag FLG is cleared by the logical value 0, and it sets Flag FLG to a logical value 1 by the completion of access. Thus, the exaggerated light of the data register SDDR which it is asynchronous and is accessed by CPU10 and the serial controller 3 by the above-mentioned regulation or control using Flag FLG can be prevented, and serial data can be exchanged asynchronously and easily between the exteriors.

[0044] Said State controller 153 receives 1-bit a reset signal/TRST, the transfer clock signal TCK, and the State control signal TMS, respectively. Let a reset signal/TRST be the reset signals which initialize the serial interface circuit 15. The transfer clock signal TCK is made into the synchronizing clock signal of the data transfer through said serial data input/output terminals TDI and TDO, and is made into the reference clock signal of the State controller 153 of operation. Moreover, the State controller 153 can recognize whether the logical value of the outputted flag FLG is 1, when said flag FLG is outputted from the serial data output terminal TDO through the shift parallel conversion register 150 with reference to the serial data output terminal TDO.

[0045] Said State controller 153 is made into the so-called state machine, makes the internal control state of the serial interface circuit 15 change in the sequence defined beforehand, and determines the transition direction at this time with the logical value of said State control signal TMS.

[0046] The transition diagram of the control state by said State controller 153 is illustrated by drawing 5. For ST1, the reset State and ST2 are [the selection State of a data register SDDR (or the status register SDSR) and ST4 of an idle state and ST3] the selection State of a command register SDIR. The direction which progresses to the following State from the current State is determined by the logical values 1 and 0 of the State control signal TMS.

[0047] It is determined by whether the flag FLG of a logical value 1 was outputted outside through the serial data output terminal TDO any of Registers SDDR and SDSR are chosen in the State ST 3. That is, the State controller 153 chooses a data register SDDR in said State ST 3 by recognizing that the flag FLG of a logical value 1 was outputted outside through the serial data output terminal TDO.

[0048] In said State ST 3, when made TMS=0 The capture State ST 30 which outputs the data of Register SDDR (SDSR) to the shift parallel conversion register 150, and the shift parallel conversion register 150 At the time of control signal TMS=0 It synchronizes with a transfer clock TCK. The shift state ST 31 which carries out a shift action, the activity State From ST32, the pause state ST 33, the activity State (reserve State for a switch of operation) ST 34, and the shift parallel conversion register 150, to Register SDDR (SDSR) (Reserve State for a switch of operation) Data It enables it to take the update State ST 35 which carries out a juxtaposition output one by one according to the logical value of the State control signal TMS. The data which incorporate the data supplied to the serial data input terminal TDI to Register SDDR through the shift parallel conversion register 150, and are set to Register SDDR (SDSR) by this can be outputted from the serial data output terminal TDO through the shift parallel conversion register 150.

[0049] For example, Flag FLG is initialized by the logical value 1 by reset of the serial interface circuit 15. If the register chosen by the State ST 3 is used as a status register SDSR in this condition and Flag FLG is read outside through the capture State ST 30 When the State controller 153 detects it and it returns to the State ST 3 shortly, in the State ST 3 concerned A data register SDDR is chosen and the data supplied from the serial data input terminal TDI can be stored in a data register SDDR through the shift state ST 31 and update State ST35 grade. Although not restricted especially, once it escapes from the update State ST 35, the register-select condition in the subsequent State ST 3 will be initialized by the status register SDSR.

[0050] In the selection State ST 4 of a command register SDIR, when made TMS=0 The command of a command register SDIR The capture State ST 40 outputted to the shift parallel conversion register 150, the shift state ST 41 of the shift parallel conversion register 150, the activity State (reserve State for a switch of operation) ST 42, the pause state ST 43, the activity State (Reserve State for a switch of operation) It enables it to take the update State ST 45 which outputs a command to a command register SDIR one by one according to the logical value of the State control signal TMS from ST44 and the shift parallel conversion register 150. By this, the command supplied to the serial data input terminal TDI can be incorporated to a command register SDIR through the shift parallel conversion register 150. In addition, an intermediate control state may be used as a dummy depending on the direction which makes a condition change.

[0051] The mode of operation of <<microcomputer, and address space>> It is the system controller which is shown by the reference mark 17 in drawing 1. A system controller 17 controls the mode of operation of a microcomputer 1 etc. The emulator mode signal ASEMD0 is typically shown as a mode signal by this example. Although not restricted especially, the reset signal of a microcomputer 1 / RST is also supplied to the system controller 17.

[0052] At the time of ASEMD 0= 0, a microcomputer 1 is made into ASE mode (evaluation mode or debug mode), and let a microcomputer 1 be body chip mode (real mode) at the time of ASEMD 0= 1 so that it may be illustrated by drawing 4.

[0053] The real mode is the mode (it is not a debug mode) in which the microcomputer 1 as an evaluation chip is operated as a real chip. It considers as the space of the user memory area (User_MEM) 110, external memory space (external memory space of a microcomputer 1), and the space for circumference modules, and let other space be reserve space (space which cannot be used) so that the address space of CPU10 at this time may be illustrated by drawing 6. Let space for circumference modules be internal register space, such as the circumference circuit 12, for example, said interruption controller, included in a real chip, and a bus controller 13, etc. Let space of said user memory area 110 be the work-piece field which consists of built-in RAM and built-in ROM, data temporary storage, and a

program field with a real chip. In the real mode, CPU10 fetches an instruction from the program field arranged in the space or external memory space of the user memory area 110, and performs this. The field of the vector table for exception handling, such as reset processing, is assigned; if CPU10 is reset in the real mode by the head of the user memory area 110 (power on reset, a hardware reset, or software reset), the reset vector arranged at the head of the vector table concerned (H'00000000) is acquired at it, the reset processing program shown by it is first performed at it, and program manipulation is advanced to it.

[0054] Said evaluation mode is a debug mode, and especially, CPU10 calls a user mode said real mode and the mode of operation which can perform the user program by which it was similarly stored in the user memory area 110, and calls breaking mode the mode of operation which can perform the debug program stored in said RAM field 111 for debugging.

[0055] The address space of CPU10 in said user mode is illustrated by drawing 7, and does not change at all to the real mode.

[0056] The address space of CPU10 in said breaking mode is illustrated by drawing 8, and emulator space (interior/exterior) is added to the user mode. Said external emulator space can be used for trace space etc. when there is an ASE terminal which is not illustrated. The field where said internal emulator space is illustrated by drawing 9 is assigned. Internal register SDIR of said serial interface circuit 15, SDSR, SDDR, the internal register 140 of the breaking controller 14, etc. are mapped by the space for circumference modules shown in drawing 6, drawing 7, and drawing 8. In the address map of drawing 9, even if R0 save-area 111A performs processing to which the value of the internal register R0 of CPU10 is temporarily evacuated, and CPU10 attaches on the occasion of a mode-of-operation switch using a register R0 on the occasion of the switch in breaking mode from a user mode, it is a field for enabling reference of the value of the register R0 at the switch time in a user mode afterwards. Hardware BURE vector field 111B is a vector storing field used for a switch of the program space of CPU10; when hardware breaking occurs. Hardware breaking is generated by said interrupt request signal brks and SDI_brk. Software breaking vector field 111C is the storing field of the vector used for a switch of the program space of CPU10, when software breaking occurs. Said R0 save area 111A, hardware breaking vector field 111B, and software breaking vector field 111C are made into a part of RAM field 111 for debugging, and are offered by RAM. SDI boot vector field 112A is the storing field of the vector which CPU10 refers to, when interrupt request signal SDI_boot is received. For example, let the vector stored in this field 112A be the information indicating start-address H'FFFC0 of the boot program field 112. This SDI boot vector field 112A is made into a part of boot program field 112, and is offered by ROM.

[0057] Boot actuation>> for the breaking mode establishment by <<SDI_boot Transition of the mode of operation between breaking mode and a user mode is explained to establishment in the breaking mode in evaluation mode (debug mode), and a list below.

[0058] The control procedure of the boot actuation for establishing said breaking mode is ****(ed) by drawing 10, and the timing chart at that time is shown in drawing 11.

[0059] In order to establish a debug mode, while making said mode signal ASEMD0 into a low level, the reset signal of a microcomputer 1 / RST, and the reset signal/TRST of the serial interface circuit 15 are made into a low level, respectively (S1). Thereby, the serial interface circuit 15 is initialized and a microcomputer 1 is made into a reset HOLD status (S2). Subsequently, after making a reset signal/RST high-level, the reset HOLD status of a microcomputer 1 is maintained by making a reset signal/TRST high-level (S3). In this condition, a microcomputer 1 is made into a debug mode and CPU10 takes the address mapping in the breaking mode shown in drawing 8. In the reset HOLD status of a microcomputer 1, actuation of CPU10, the interruption controller 12, and the serial interface circuit 15 is enabled.

[0060] Synchronizing with a transfer clock TCK, the State control signal TMS changes to necessary in this condition, if command register SDIR supply of the boot command is carried out from the serial data input terminal TDI, (S4) and its boot command will be decoded by the command decoder 154, and by it, a

command decoder 154 interrupts and outputs boot interrupt request signal SDI_boot to a controller 12 (S5).

[0061] The interruption controller 12 will supply the factor information and the interrupt signal corresponding to the interruption concerned to CPU10, if the boot interrupt request signal SDI_boot is received. A reset HOLD status is canceled by the interrupt signal, CPU10 acquires the vector address in hardware based on the interruption factor information, and CPU10 acquires the vector of SDI boot vector field 112A shown in drawing 9 by this. If this example is followed, said vector will show start-address H'FFFFFC0 of the boot program field 112, and CPU10 will start activation of the boot program stored in the boot program field 112 by this (S6).

[0062] A boot program is considered as the program which carries out transfer control of the information supplied through the serial data input terminal TDI to the RAM field 111 for debugging. The information on the vector table which this uses for the debug program supplied to the serial data input terminal TDI through the serial interface circuit 15 or it downloads to the RAM field 111 for debugging (S7).

[0063] If this example is followed, the branch instruction which branches the instruction-execution condition of CPU10 to start-address H'FFFF800 of said RAM field 111 for debugging is included in the last of a boot program. Thereby, following activation of a boot program, the instruction execution of CPU10 branches to the debug program of the RAM field 111 for debugging (S8), and the condition that CPU10 can perform a debug program changes (S9). Thus, a microcomputer 1 establishes the breaking mode in evaluation mode.

[0064] In drawing 11, time of day t1 shows the timing by which said reset HOLD status is attained. If made a reset HOLD status, it will be made breaking mode to the timing of time of day t2. If a reset signal/RST is made high-level at time of day t3, the reset condition of the serial interface circuit 15 will be canceled by changing a reset signal/TRST high-level after that. However, the reset HOLD status of CPU10 is maintained. If said boot command is published at time of day t4, the reset HOLD status of CPU10 is canceled by interruption, and CPU10 will execute a boot program and will download the data inputted from the serial data input terminal TDI to the RAM field 111 for debugging. If activation of a boot program is completed at time of day t5, the instruction-execution condition of CPU10 will branch to the debug program downloaded to the RAM field 111 for debugging as above-mentioned.

[0065] the contents of the debug program are alike, respond to the structure of a system which applies a microcomputer 1, and a user determines them as arbitration.

[0066] In this example, the transfer control program for downloading a user program (program for debugging) to the program memory space of the user memory area 110 through said serial interface circuit 15 can be arranged in the head of the RAM field 111 for debugging. And the program for outputting the condition inside a microcomputer outside through said serial interface circuit 15, the program for setting breaking conditions as said breaking controller 14 through said serial interface circuit 15, etc. can be included in said debug program by making such a transfer control program into the start. Or the program which enables actuation for choosing such a program can also be arranged. Moreover, if the instruction which performs actuation for branching the address of the program storage area of a request of the instruction-execution condition of CPU10 of said user memory area 111 and external memory is arranged, it is also possible to shift to a user mode from breaking mode.

[0067] Breaking>> by <<SDI_brk The transition control procedure to the breaking mode by interrupt request signal SDI_brk is shown in drawing 12, and the timing chart at that time is shown in drawing 13.

[0068] In a user mode, the breaking controller 14 and the serial interface circuit 15 cannot be accessed. In a user mode, if the breaking conditions set as the breaking controller 14 in breaking mode are satisfied, a microcomputer 1 will shift to breaking mode with the breaking interrupt request signal brks. In the user mode, reception of the breaking interrupt request by SDI_brk and the interrupt request by SDI_int mentioned later is enabled. The breaking interrupt request by SDI_brk is not received in the real mode.

[0069] For starting breaking by SDI breaking interrupt request signal SDI_{brk}, a reset signal/TRST is first made into a low level, the serial interface circuit 15 is initialized (S10), and a reset signal/TRST is returned high-level after that (S11). And if a breaking command is supplied from the serial data input terminal TDI (S12), the command will be decoded by the command decoder 154, consequently interrupt request signal SDI_{brk} will interrupt, and it will be outputted to a controller 12 (S13).

[0070] The interruption controller 12 will supply the factor information and the interrupt signal corresponding to the interruption concerned to CPU10, if the breaking interrupt request is received. CPU10 acquires the vector address in hardware based on the interruption factor information, and CPU10 acquires the vector of hardware breaking vector field 111B shown in drawing 9 by this. Some fields of the RAM field 111 for debugging are assigned, this hardware breaking vector field 111B responds to the vector set as it, processing of CPU10 branches to predetermined processing of a debug program (S14), and CPU10 performs the program of a branching place and performs emulation actuation (S15). For example, actuation which sets breaking conditions as the breaking controller 14, or reads the value of the internal register of CPU10 outside through the serial interface circuit 15 through the serial interface circuit 15 is performed.

[0071] In drawing 13, when a microcomputer 1 is a user mode (the case where it is in breaking mode is not excepted), the serial interface circuit 15 is reset by a reset signal/TRST at time of day t1. The circuit of others of a microcomputer 1 is not reset by it. And after canceling reset of the serial interface circuit 15, a breaking command is supplied to the serial interface circuit 15 from the serial data input terminal TDI at time of day t2. By this, as above-mentioned, a microcomputer 1 changes in breaking mode by breaking interrupt request signal SDI_{brk}, and CPU10 branches to predetermined processing of the debug program of the RAM field 111 for debugging, for example, the data for new breaking conditioning are received through the serial interface circuit 15.

[0072] Interruption>> by <<SDI_int The interrupt control procedure by interrupt request signal SDI_{int} is shown in drawing 14, and the timing chart at that time is shown in drawing 15.

[0073] Reception of interruption by interrupt request signal SDI_{int} in the real mode or a user mode is enabled (in breaking mode, reception is impossible). It is because the vector table referred to by this interruption must be formed in the user memory area 110.

[0074] For starting interruption by SDI interrupt request signal SDI_{int}, a reset signal/TRST is first made into a low level, the serial interface circuit 15 is initialized (S20), and a reset signal/TRST is returned high-level after that (S21). And if an interruption command is supplied from the serial data input terminal TDI (S22), the command will be decoded by the command decoder 154, consequently interrupt request signal SDI_{int} will interrupt, and it will be outputted to a controller 12 (S23).

[0075] The interruption controller 12 will supply the factor information and the interrupt signal corresponding to the interruption concerned to CPU10, if the interrupt request is received. CPU10 acquires the vector address based on the interruption factor information, and CPU10 acquires a vector from the user memory area 110 by this. The vector is determined as arbitration according to the user program, it responds to the vector, processing of CPU10 branches to predetermined processing of a user program (S24), and CPU10 performs the user program of a branching place (S25).

[0076] In drawing 15, when a microcomputer 1 is a user mode (the case where it is in body chip mode is not excepted), the serial interface circuit 15 is reset by a reset signal/TRST at time of day t1. The circuit of others of a microcomputer 1 is not reset by it. And after canceling reset of the serial interface circuit 15, at time of day t2, the serial interface circuit 15 is interrupted from the serial data input terminal TDI, and a command is supplied. It branches by this to the processing as which, as for processing of CPU10, interruption was required by interrupt request signal SDI_{int} as above-mentioned.

[0077] The overall example of a system configuration in the case of debugging the target system which applied said microcomputer 1 is shown in drawing 16. The configuration shown in this drawing corresponds to drawing 2. The interface of the target system 5 and a debug tool 4 is performed through the serial controller 3 and said serial interface circuit 15. The debug tool 4 makes an example the

portable personal computer which can perform an emulation program. The serial controller 3 incorporates and controls the serial data which carries out serial supply of the data and is outputted to the serial data input terminal TDI from the serial data output terminal TDO while controlling said signal/TRST, and TCK and TMS according to directions of the information transfer from a debug tool 4. Although the protocol of serial transmission is based on the so-called JTAG used for the test of the semiconductor integrated circuit of a large number mounted in the circuit board in the above-mentioned example, it is not limited to it.

[0078] In drawing 16 , the external connection connector 51 is arranged for that for connecting the serial interface circuit 15 of said microcomputer 1 with the serial controller 3 at the circuit board 55 of the target system 5. this external connection connector 51 corresponds to five signals (/TRST, TCK, TMS, TDI, TDO) — sufficient — a big area is not needed on the circuit board. It is suitable for the small target system like a pocket communication terminal or a pocket device. This external connection connector 51 is connected to the serial controller 3 through the interface cable 30.

[0079] For example, since the number of a signal line required for the interface of the target system 5 and a debug tool is lessened as much as possible also when bringing the system debug or software debugging close to a real operating environment as much as possible when the target system 5 is a pocket communication terminal like a portable telephone, and carrying out outdoors, it is rich in mobility [say / that it can respond also to such a use gestalt immediately].

[0080] As explained above, a microcomputer 1 is made to perform the user program made applicable to debugging in the target system 5 a software WEDE bag or when carrying out system debug, and it debugs by following the activation locus of the user program concerned, referring to the information acquired in the process externally. Some which hold the trace information for following said user program activation locus are in the interior of said breaking controller 14. In order to refer to the information which appears in the activation process of such a user program externally or to set up the operating condition of a user program etc., it is needed to make a microcomputer 1 perform a different debug program from a user program. Said boot program is a program of CPU10 ** for downloading such a debug program to a RAM field 111 for debugging which is different in the storing field 110 of a user program of operation. By creating beforehand the debug program needed according to the configuration of the target system 5 to necessary, and publishing said SDI boot command in the serial interface circuit 15 from a debug tool 4, a microcomputer 1 starts a boot program and downloads the debug program supplied to the serial interface circuit 15 to the RAM field 111 for debugging. Therefore, when referring to the information which appears in the activation process of a user program externally on the occasion of debugging or setting up the operating condition of a user program etc., the contents of the actuation will be performed according to the downloaded contents of the debug program. Thereby, any debugging actuation can be performed according to the contents of the program, and the variability of debugging actuation can be guaranteed. The debugging actuation by the microcomputer 1 is not restricted by the class of command receivable in the serial interface circuit 15.

[0081] The serial interface circuit 15 is used for an interface with a debug tool 4, the number of the terminals for interfaces for debugging is lessened, and it can be compatible in the variability of debugging, and good user-friendliness under such a limit.

[0082] Also when bringing system debug or software debugging close to a real operating environment as much as possible and performing it to it outdoors to the target system 5 like a pocket communication terminal or a pocket device, there are few terminals for interfaces for debugging, and since the variability and the good user-friendliness of debugging are also guaranteed, it is rich in mobility [say / that it can respond immediately].

[0083] the predetermined address of said debug program to which the instruction execution of CPU10 was transmitted to said RAM field 111 for debugging, for example, the instruction which it is [instruction] alike and branches H'FFFF800, can be included in the last at said boot program. The downloaded debug program can be started automatically by this, and first stage-control action can be

started immediately.

[0084] If the download of a user program made applicable to debugging as first stage-actuation is assumed, said internal memory 11 has the rewritable user memory area 110, and said debug program can contain the transfer control program which makes a user program etc. transmit to said user memory area 110.

[0085] since a user program can be arranged to the user memory area 110 of the microcomputer 1 interior on the occasion of debugging — actually — ** — a user program can be performed with the same working speed for debugging.

[0086] Said serial interface circuit 15 is based on the SDI breaking command supplied from the outside, and is a SDI interrupt request signal (outputting SDI_{brk}, said interrupt controller 12 answers the interrupt request signal, and gives the interrupt control information which makes CPU10 execute the predetermined instruction of said debug program to CPU10.) to said interrupt controller 12. Therefore, actuation (breaking actuation) which switches the program of CPU10 of operation to said debug program can be performed to arbitration through the serial interface circuit 15 on the way of [of a user program / activation] etc. from host equipment 4.

[0087] At this time, hardware breaking vector field 111B used in order that CPU10 which received the interrupt control information based on a SDI-breaking command may acquire the predetermined instruction address of said debug program is assigned to said RAM field 111 for debugging. According to said debug program, since this field is a field which can rewrite CPU10, it can also determine as arbitration the debugging processing which you want to perform by issue of such a breaking command.

[0088] Based on the SDI interruption command supplied from the outside, said serial interface circuit 15 outputs SDI interrupt request signal SDI_{int} to said interrupt controller 12, and said interrupt controller 12 answers the interrupt request signal, and it gives the interrupt control information which makes CPU10 execute the program of the user memory area 1110 or other user rooms to CPU10. Thereby, arbitration can be interrupted through the serial interface circuit 15 on the way of [of a user program / activation] etc. from host equipment 4.

[0089] When outputting and inputting information asynchronously and serially with the exterior of a microcomputer 1 through the serial interface circuit 15, by permitting access of the data register SDDR by CPU10 according to the condition of Flag FLG, the exaggerated light of the data register SDDR which it is asynchronous and is accessed by the exterior and CPU10 of a microcomputer 1 can be prevented, and data can be serially exchanged asynchronously and easily between the exteriors.

[0090] The State controller 153 detects the condition that said flag FLG made into the logical value 1 was outputted to the exterior of a microcomputer 1. While choosing said data register SDDR from the exterior of a microcomputer 1 accessible then Since modification of said flag FLG to a logical value 0 is enabled with the completion of access of the data register SDDR by the exterior, an exclusive signal line cannot be used for outputting the information on Flag FLG to the exterior of a microcomputer 1, but the signal line for serial data I/O can be used.

[0091] Although invention made by this invention person above was concretely explained based on the operation gestalt, it cannot be overemphasized that it can change variously in the range which this invention is not limited to it and does not deviate from the summary:

[0092] For example, a data processor is not limited to a microcomputer, but is large in the logic LSI of the microcontroller which specialized in the DIJITA signal processor which specialized in the digital-signal-processing operation, a microprocessor, and appliance control, and can be applied to it. The circuit module built in these data processors is not limited to the above-mentioned example.

[0093]

[Effect of the Invention] It will be as follows if the effectiveness acquired by the typical thing among invention indicated in this application is explained briefly.

[0094] That is, any debugging actuation can be performed according to the contents of the program for debugging control downloaded by the boot program, and the variability of debugging actuation can be

guaranteed. The debugging actuation by the data processor is not restricted by the class of command receivable [with a serial interface means]. [0095] A serial interface means is used for an interface with a host system; the number of the terminals for interfaces for debugging is lessened, and it can be compatible in the variability of debugging, and good user-friendliness under such a limit.

[0096] Also when bringing system debug or software debugging close to a real operating environment as much as possible and performing it to it outdoors to the target system like a pocket communication terminal or a pocket device, there are few terminals for interfaces for debugging, and since the variability and the good user-friendliness of debugging are also guaranteed, mobility [say / that it can respond immediately] can be acquired.

[0097] Moreover, by including the instruction which branches the instruction execution of a central processing unit to the predetermined address of the program for debugging control in the last of said boot program, the downloaded program for debugging control can be started automatically, and first stage-control action can be started immediately.

[0098] arranging a user program in the memory inside a data processor on the occasion of debugging — actually — ** — a user program can be performed with the same working speed for debugging.

[0099] The interrupt request which answers the 2nd command (SDI breaking command) supplied to a serial interface means from the exterior can perform at arbitration actuation (breaking actuation) which switches the program of a central processing unit of operation to said control program for debugging to a central processing unit through a serial interface means on the way of [of a user program / activation] etc. by performing a predetermined instruction of the program for debugging control from a host equipment side.

[0100] A central processing unit can be made to generate [by the interrupt request which answers the 3rd command (SDI interruption command) supplied to a serial interface means from the exterior] an interruption in arbitration through a serial interface means on the way of [of a user program / activation] etc. from a host equipment side by enabling branching of a central processing unit to the predetermined program of user room.

[0101] When outputting and inputting information asynchronously and serially with the exterior of a data processor through a serial interface means, by permitting access of the data register by the central processing unit according to the condition of a flag (FLG), the exaggerated light of the data register (SDDR) asynchronous and accessed by the exterior and the central processing unit of a data processor, can be prevented, and data can be serially exchanged asynchronously and easily between the exteriors.

[Translation done.]

* NOTICES *
JPO and NCIP are not responsible for any damages caused by the use of this translation.

1. This document has been translated by computer. So the translation may not reflect the original precisely.
 2. **** shows the word which can not be translated.
 3. In the drawings, any words are not translated.
-

DESCRIPTION OF DRAWINGS

[Brief Description of the Drawings]

[Drawing 1] It is the block diagram of the microcomputer concerning an example of this invention.

[Drawing 2] It is the block diagram showing an example in the condition of having carried out the interface of the target system which applied the microcomputer of drawing 1 to the debug tool.

[Drawing 3] It is the explanatory view showing an example of the command register contained in a serial interface circuit.

[Drawing 4] It is the explanatory view showing an example of the mode of operation of the microcomputer set up by emulator mode signal ASEMOD.

[Drawing 5] It is the transition diagram of the state-transition control by the State controller contained in a serial interface circuit.

[Drawing 6] It is the explanatory view showing the address space of CPU in the real mode.

[Drawing 7] It is the explanatory view showing the address space of CPU in a user mode.

[Drawing 8] It is the explanatory view showing the address space of CPU in breaking mode.

[Drawing 9] It is the explanatory view showing the detail of the internal emulator space assigned in breaking mode.

[Drawing 10] It is an example flow chart which shows the control procedure of the boot actuation by interrupt request signal SDI_boot for establishing breaking mode.

[Drawing 11] It is the timing chart which shows an example of the boot actuation by interrupt request signal SDI_boot for establishing breaking mode.

[Drawing 12] It is an example flow chart which shows the control procedure of the transition to the breaking mode by interrupt request signal SDI_brk.

[Drawing 13] It is the timing chart which shows an example of the transition actuation to the breaking mode by interrupt request signal SDI_brk.

[Drawing 14] It is an example flow chart which shows the control procedure of interruption by interrupt request signal SDI_int.

[Drawing 15] It is the timing chart which shows an example of the control action of interruption by interrupt request signal SDI_int.

[Drawing 16] It is the perspective view showing an example in the condition of having carried out the interface of the target system which applied the microcomputer of drawing 1 to the debug tool.

[Description of Notations]

1 Microcomputer

2 External Memory

3 Serial Controller

4 Debug Tool

5 Target System

50 Circuit Board

51 External Connection Connector

10 CPU (Central Processing Unit)

110 User Memory Area

111 RAM Field for Debugging

111B Hardware breaking vector field

112 Boot Program Field

112A SDI boot vector field

11 Internal Memory

12 Interruption Controller

13 Bus Controller

14 Breaking Controller

15 Serial Interface Circuit

/TRST Reset signal

TCK Transfer clock signal

TMS State control signal

TDI Serial data input terminal
TDO Serial data output terminal
150 Shift Parallel Conversion Register
153 State Controller
154 Command Decoder
SDIR Command register
SDSR base — status register
FLG Flag
SDDR Data register
SDI_boot SDI boot interrupt request signal
SDI_int SDI interrupt request signal
SDI_brk SDI breaking interrupt request signal
16D Internal data bus
16A Internal address bus

[Translation done.]

(19)日本国特許庁 (JP)

(12) 公開特許公報 (A)

(11)特許出願公開番号

特開平9-319727

(43)公開日 平成9年(1997)12月12日

(51)Int.Cl. ^a	識別記号	序内整理番号	F I	技術表示箇所
G 06 F 15/78	510		G 06 F 15/78	510K
11/22	340		11/22	340A
11/28			11/28	L
13/38	350		13/38	350

審査請求 未請求 請求項の数11 O.L (全18頁)

(21)出願番号 特願平8-138111

(22)出願日 平成8年(1996)5月31日

(71)出願人 000005108

株式会社日立製作所

東京都千代田区神田駿河台四丁目6番地

(72)発明者 長谷川 博宣

東京都小平市上水本町5丁目20番1号 株式会社日立製作所半導体事業部内

(72)発明者 佐々木 宏幸

東京都小平市上水本町5丁目20番1号 株式会社日立製作所半導体事業部内

(72)発明者 浦口 正彦

東京都小平市上水本町5丁目20番1号 株式会社日立製作所半導体事業部内

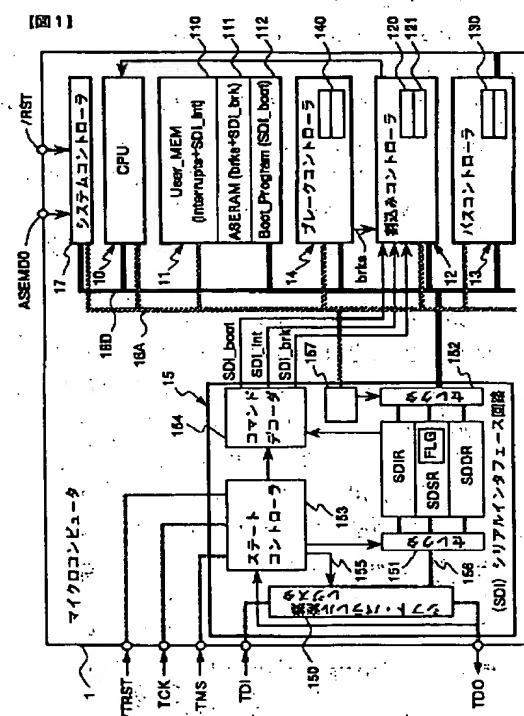
(74)代理人 弁理士 玉村 静世

(54)【発明の名称】 データプロセッサ及びデータ処理システム

(57)【要約】

【課題】 デバッグのための信号端子を少なくし、デバッグ動作を容易に可変させることができるデータプロセッサを提供する。

【解決手段】 データプロセッサ(1)は、CPU(10)にシリアルインターフェース回路(15)からデバッグ用RAM領域(111)へデバッグプログラムを転送させるブートプログラムを格納したROM(112)を備える。シリアルインターフェース回路は、外部からSDIブートコマンドが供給されると、割込みコントローラ(12)にSDI割り込み要求信号(SDI_boot)を出力し、CPUに前記ブートプログラムを実行させる。デバッグ動作はダウンロードされたデバッグプログラムの内容に従って可変であり、デバッグの際の情報のやりとりもシリアルに行われる。



(2)

2

【特許請求の範囲】

【請求項 1】 中央処理装置、この中央処理装置によってアクセス可能な記憶手段、前記中央処理装置への割込みを制御する割込みコントローラ、外部との間でシリアルに情報の入出力を行うシリアルインタフェース手段、及びそれらを接続するための内部バスを含んで半導体集積回路化されて成るデータプロセッサであって、前記記憶手段は、書き換え可能な第1の記憶領域と、前記中央処理装置に前記シリアルインタフェース手段から前記第1の記憶領域へ情報を転送させるブートプログラムを保有する第2の記憶領域とを含み、

前記シリアルインタフェース手段は、外部から供給された第1のコマンドの解読結果に基づいて前記割込みコントローラに第1の割込み要求信号を出力し、

前記割込みコントローラは、前記第1の割込み要求信号に応答して、中央処理装置に、前記ブートプログラムを実行させる第1の割込み制御情報を与えるものであることを特徴とするデータプロセッサ。

【請求項 2】 前記第2の記憶領域は更に、前記第1の割り込み制御情報を受け取った中央処理装置が前記ブートプログラムの先頭の命令アドレスを取得するために利用する第1のベクタを含んで成るものであることを特徴とする請求項1記載のデータプロセッサ。

【請求項 3】 前記ブートプログラムによって第1の記憶領域に転送される情報は、デバッグ制御用プログラムであることを特徴とする請求項1又は2記載のデータプロセッサ。

【請求項 4】 前記ブートプログラムは、その最後に、中央処理装置の命令実行を、前記第1の記憶領域に転送された前記デバッグ制御用プログラムの所定のアドレスに分岐させる命令を含んで成るものであることを特徴とする請求項3記載のデータプロセッサ。

【請求項 5】 前記記憶手段は書き換え可能な第3の記憶領域を有し、前記デバッグ制御用プログラムは、前記シリアルインタフェース手段から前記第3の記憶領域へデータを転送させる転送制御プログラムを含んで成るものであることを特徴とする請求項4記載のデータプロセッサ。

【請求項 6】 前記シリアルインタフェース手段は、外部から供給された第2のコマンドに基づいて、前記割込みコントローラに第2の割込み要求信号を出力し、

前記割込みコントローラは、前記第2の割込み要求信号に応答して、中央処理装置に、前記第1の記憶領域のデバッグ制御用プログラムの所定の命令を実行させる第2の割込み制御情報を中央処理装置に与えるものであることを特徴とする請求項3乃至5の何れか1項記載のデータプロセッサ。

【請求項 7】 前記第1の記憶領域は更に、前記第2の割り込み制御情報を受け取った中央処理装置が前記デバッグ制御用プログラムの所定の命令アドレスを取得する

ために利用する第2のベクタを含んで成るものであることを特徴とする請求項6記載のデータプロセッサ。

【請求項 8】 前記シリアルインタフェース手段は、外部から供給された第3のコマンドに基づいて、前記割込みコントローラに第3の割込み要求信号を出力し、前記割込みコントローラは、前記第3の割込み要求信号に応答して、中央処理装置に、前記第3の記憶領域又はその他のユーザメモリ空間のプログラムを実行させる第3の割込み制御情報を中央処理装置に与えるものであることを特徴とする請求項6又は7記載のデータプロセッサ。

【請求項 9】 中央処理装置、この中央処理装置によってアクセス可能な記憶手段、及び前記中央処理装置への割込みを制御する割込みコントローラを含んで半導体集積回路化されて成るデータプロセッサにおいて、データプロセッサの外部と前記中央処理装置との双方から夫々操作可能にされたフラグ、データプロセッサの外部と前記中央処理装置の双方から夫々アクセス可能にされたデータレジスタ、及び外部から供給されるコマンドの解読結果に応じた割込み要求信号を前記割込みコントローラに与えるコマンドデコーダを含み、データプロセッサの外部と非同期でシリアルに情報の入出力を行うシリアルインタフェース手段を設け、

前記記憶手段は、書き換え可能な第1の記憶領域と、前記中央処理装置に前記シリアルインタフェース手段のデータレジスタから前記第1の記憶領域へ情報を転送させるブートプログラムを保有する第2の記憶領域とを含み、

前記割込みコントローラは、前記割込み要求の中の第1の割込み要求信号に応答して、中央処理装置に、前記ブートプログラムを実行させる第1の割込み制御情報を与えるものであり、

前記ブートプログラムを実行する中央処理装置は、前記フラグが第1の状態であることを条件に前記データレジスタをアクセスし、前記データレジスタに対するアクセス完了に伴って前記フラグを第2の状態に変更するものであることを特徴とするデータプロセッサ。

【請求項 10】 第2の状態にされた前記フラグがデータプロセッサの外部に出力された状態を検出して、前記データレジスタをデータプロセッサの外部からアクセス可能に制御すると共に、データプロセッサによるデータレジスタのアクセス完了に伴って前記フラグを第1の状態に変更可能にするステートコントローラを更に備えて成ることを特徴とする請求項9記載のデータプロセッサ。

【請求項 11】 請求項1乃至10の何れか1項記載のデータプロセッサと、このデータプロセッサによってアクセス制御される回路を、回路基板に実装して成るデータ処理装置であって、前記回路基板には、前記データプロセッサに含まれるシリアルインタフェース手段に接続

(3)

3

された外部接続コネクタを設けて成るものであることを特徴とするデータ処理システム。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】本発明は、マイクロコンピュータ、ディジタル・シグナル・プロセッサ、マイクロコントローラ、マイクロプロセッサなどのデータプロセッサ、特にエミュレーション等のデバッグ支援機能を有するデータプロセッサに関し、例えば、マイクロコンピュータ応用システムを屋外などの実使用環境の下でデバッグを行う条件のものに適用して有効な技術に関するものである。

【0002】

【従来の技術】マイクロコンピュータ応用機器（ターゲットシステムとも称する）のデバッグにはデバッグ支援機能を有する評価用のマイクロコンピュータを用いることができる。この評価用のマイクロコンピュータをデバッグ用のホストシステムとインターフェースさせるために、当該マイクロコンピュータとターゲットシステムとのインターフェースと同等のデータ、アドレス及び制御信号を並列的に伝達し得る多数の信号線を用いることができる。しかしながら、ターゲットシステムが携帯通信端末のような小型システムの場合、その回路基板にデバッグ用の多数の信号線を接続するための端子若しくはコネクタを設けることは、実質的に不可能な場合が多い。

【0003】そこで、デバッグ用のホストシステムとのインターフェースにシリアルインターフェースを用いる技術が、1989年12月にIEEE発行の「DSP Floating Point Dual-Port Processor User's Manual SECTION10 ON-CHIP EMULATOR」によって提供されている。

【0004】これに記載された技術は、ディジタル・シグナル・プロセッサに、デバッグ用のコマンドデコーダ、ブレークポイント制御回路、及びクロック信号に同期してデータをシリアルに入出力するデバッグ用のシリアルインターフェースを内蔵させ、このシリアルインターフェースを介して、デバッグ用のコマンドを外部から与え、そのコマンドをコマンドデコーダで解読し、その解読結果に応じて、内部レジスタのリード／ライトやブレークポイント制御を行うようになっている。

【0005】

【発明が解決しようとする課題】しかしながら、上記従来技術にあっては、デバッグ用のコマンドの機能が予め固定的に決定されているため、デバッグに当たりデータプロセッサに模擬的に設定可能な内部状態や、デバッグのために参照できるデータプロセッサの内部状態などが制限される。ディジタル・シグナル・プロセッサなどのデータプロセッサは種々のデータ処理システムに汎用的に適用されることを考慮すれば、そのような制限を受けるデバッグ支援機能では今一つ使い勝手が良好ではない。換言すれば、デバッグのための動作に可変性がない。

(3)

4

い。

【0006】また、前記シリアルインターフェースを採用することによって、デバッグ用ホスト装置との接続端子を減らすことができるが、その信号数を最小限にするには、データプロセッサの外部と非同期でシリアルに情報の入出力をを行う際の、ハンドシェーク制御の手法についても考慮する必要性が見いだされた。

【0007】本発明の目的は、デバッグのための信号端子を極力少なくできるデータプロセッサを提供することにある。

【0008】本発明の別の目的は、デバッグ動作を容易に可変させることができるデータプロセッサを提供することにある。

【0009】本発明の更に別の目的は、実機（製品システム）の実際の利用環境に近い環境でデバッグを可能にするデータプロセッサ、及び当該データプロセッサを用いたデータ処理システムを提供することにある。

【0010】本発明の前記並びにその他の目的と新規な特徴は本明細書の記述及び添付図面から明らかになるであろう。

【0011】

【課題を解決するための手段】本願において開示される発明のうち代表的なものの概要を簡単に説明すれば下記の通りである。

【0012】すなわち、データプロセッサ（1）は、中央処理装置（10）、この中央処理装置によってアクセス可能な記憶手段（11）、前記中央処理装置への割込みを制御する割込みコントローラ（12）、外部との間でシリアルに情報の入出力をを行うシリアルインターフェース手段（15）、及びそれらを接続するための内部バス（16）を含んで半導体集積回路化されて成る。前記前記記憶手段（11）は、書き換え可能な第1の記憶領域（111）と、前記中央処理装置に前記シリアルインターフェース手段から前記第1の記憶領域へ情報を転送させるブートプログラムを保有する第2の記憶領域（112）とを含み、前記シリアルインターフェース手段（15）は、外部から供給された第1のコマンド（SDIポートコマンド）の解読結果に基づいて前記割込みコントローラに第1の割込み要求信号（SDI_boot）を出力し、前記割込みコントローラ（12）は、前記第1の割込み要求信号に応答して、中央処理装置に、前記ブートプログラムを実行させる第1の割込み制御情報を与える。

【0013】前記データプロセッサの適用システム（ターゲットシステム）をソフトウェーブデバッグ若しくはシステムデバッグする場合、デバッグ対象とされるユーザプログラムをデータプロセッサに実行させ、その過程で得られた情報を外部で参照しながら、当該ユーザプログラムの実行軌跡をたどって、デバッグを行う。このとき、そのようなユーザプログラムの実行過程で現れる情報を外部で参照したり、あるいは、ユーザプログラムの動作

50

(4)

6

条件等を設定するには、ユーザプログラムとは異なるデバグ制御用プログラムをデータプロセッサに実行させることが必要になる。前記ブートプログラムは、そのようなデバグ制御用プログラムを、ユーザプログラムの格納領域とは異なる領域にダウンロードするための中央処理装置の動作プログラムである。デバグ対象システムの構成に応じて必要とされるデバグ制御用プログラムを予め所要に作成しておき、前記第1のコマンドをホスト装置からシリアルインタフェース手段に発行することにより、データプロセッサはブートプログラムを起動し、シリアルインタフェース手段に供給されるデバグ制御用プログラムを、第1の記憶領域にダウンロードする。したがって、デバグに際して、ユーザプログラムの実行過程で現れる情報を外部で参照したり、あるいは、ユーザプログラムの動作条件等を設定するとき、その動作の内容は、ダウンロードされたデバグ制御用プログラムの内容に従って行われることになる。これにより、そのプログラムの内容次第でどのようなデバグ動作も行うことができ、デバグ動作の可変性を保証する事ができる。データプロセッサによるデバグ動作が、シリアルインタフェース手段で受け付け可能なコマンドの種類によって制限されることはない。

【0014】ホストシステムとのインターフェースにはシリアルインタフェース手段を利用し、デバグの為のインターフェース用端子の数を少なくしており、そのような制限の下においても、デバグの可変性と、良好な使い勝手とを両立することができる。

【0015】携帯通信端末若しくは携帯機器のようなターゲットシステムに対し、システムデバグ若しくはソフトウェアデバグを、実使用環境に極力近付けて屋外で行う様な場合にも、デバグの為のインターフェース用端子の数が少なく、また、デバグの可変性や良好な使い勝手も保証されているので、即座に対応することができるという、機動性に富む。

【0016】上記手段の具体的な態様として、前記第2の記憶領域は更に、前記第1の割り込み制御情報を受け取った中央処理装置が前記ブートプログラムの先頭の命令アドレスを取得するために利用する第1のベクタを格納する領域(112A)を含むことができる。

【0017】また、前記ブートプログラムには、その後に、中央処理装置の命令実行を、前記第1の記憶領域に転送された前記デバグ制御用プログラムの所定のアドレスに分岐させる命令を含めることができる。これにより、ダウンロードされたデバグ制御用プログラムを自動的に起動して、初期的な制御動作を即座に開始することができる。

【0018】初期的な動作としてデバグ対象とされるユーザプログラムのダウンロードを想定すると、前記記憶手段は書き換え可能な第3の記憶領域(110)を有し、前記デバグ制御用プログラムは、前記第3の記憶

領域へユーザプログラムなどを転送させる転送制御プログラムを含むことができる。

【0019】デバグに際してユーザプログラムをデータプロセッサ内部のメモリに配置することができるのと、実際と同じ動作速度でデバグの為にユーザプログラムを実行することができる。

【0020】ユーザプログラムの実行途上等で、中央処理装置の動作プログラムを前記デバグ用制御プログラムに切り換える動作(ブレーク動作)をホスト装置側からシリアルインタフェース手段を介して任意に行えるようにするには、前記シリアルインタフェース手段は、外部から供給された第2のコマンド(SDIブレークコマンド)に基づいて、前記割込みコントローラに第2の割込み要求信号(SDI_brk)を出力し、前記割込みコントローラは、前記第2の割込み要求信号に応答して、中央処理装置に、前記第1の記憶領域のデバグ制御用プログラムの所定の命令を実行させる第2の割込み制御情報を中央処理装置に与えるようにすればよい。

【0021】このとき、前記第1の記憶領域には、前記第2の割り込み制御情報を受け取った中央処理装置が前記デバグ制御用プログラムの所定の命令アドレスを取得するために利用する第2のベクタを格納する領域(111B)を割り当ておくことができる。この領域は前記デバグ制御用プログラムに従って中央処理装置が書き換え可能な領域であるから、そのようなブレークコマンドの発行によって実行させたいデバグ処理も任意に決定することができる。

【0022】ユーザプログラムの実行途上等で、ホスト装置側からシリアルインタフェース手段を介して任意に割り込みを行えるようにするには、前記シリアルインタフェース手段は、外部から供給された第3のコマンド(SDI割り込みコマンド)に基づいて、前記割込みコントローラに第3の割込み要求信号(SDI_int)を出力し、前記割込みコントローラは、前記第3の割込み要求信号に応答して、中央処理装置に、前記第3の記憶領域又はその他のユーザメモリ空間のプログラムを実行させる第3の割込み制御情報を中央処理装置に与えるようにすることができる。

【0023】デバグ動作のための情報をシリアルインタフェース手段を介してホスト装置とやりとりするデータプロセッサの構成において、シリアルインタフェース手段に着目した別の観点によるデータプロセッサ(1)は、中央処理装置(10)、この中央処理装置によってアクセス可能な記憶手段(11)、前記中央処理装置への割込みを制御する割込みコントローラ(12)及びシリアルインタフェース手段(15)を含んで半導体集積回路化されている。前記シリアルインタフェース手段は、データプロセッサの外部と前記中央処理装置との双方から夫々操作可能にされたフラグ(FLG)、データプロセッサの外部と前記中央処理装置の双方から夫々ア

(5)

7 クセス可能にされたデータレジスタ（SDDR）、及び外部から供給されるコマンドの解読結果に応じた割込み要求信号を前記割込みコントローラに与えるコマンドデコーダ（154）を含み、データプロセッサの外部と非同期でシリアルに情報の入出力を行う。このとき、前記記憶手段は、書き換え可能な第1の記憶領域（111）と、前記中央処理装置に前記シリアルインタフェース手段のデータレジスタから前記第1の記憶領域へ情報を転送させるブートプログラムを保有する第2の記憶領域

（112）とを含み、前記割込みコントローラは、前記割込み要求の中の第1の割込み要求信号（SDI_boot）に応答して、中央処理装置に、前記ブートプログラムを実行させる第1の割込み制御情報を与える。前記ブートプログラムを実行する中央処理装置は、シリアルインタフェース手段との間でデータをやりとりするとき、前記フラグ（FLG）が第1の状態であることを条件に前記データレジスタをアクセスし、前記データレジスタに対するアクセス完了に伴って前記フラグを第2の状態に変更する。

【0024】このように、シリアルインタフェース手段を介して、データプロセッサの外部と非同期でシリアルに情報の入出力を行うとき、フラグ（FLG）の状態に応じて中央処理装置によるデータレジスタのアクセスを許容することにより、データプロセッサの外部と中央処理装置によって非同期でアクセスされるデータレジスタ（SDDR）のオーバライトを防いで、外部との間で、非同期で容易にデータをシリアルにやりとりすることができる。

【0025】データプロセッサの外部においてはフラグが第2の状態にされているとき、データレジスタをアクセスし、アクセス完了と共にフラグを第1の状態に反転させねばよい。

【0026】データプロセッサの外部へフラグの情報を出力するのに専用信号線を用いず、シリアルデータ入出力のための信号線を用いるようにするには、第2の状態にされた前記フラグがデータプロセッサの外部に出力された状態を検出して、そのとき前記データレジスタをデータプロセッサの外部からアクセス可能に制御すると共に、データプロセッサによるデータレジスタのアクセス完了に伴って前記フラグを第1の状態に変更可能にするステートコントローラ（153）を採用すればよい。

【0027】前述のデータプロセッサを適用して構成されるデータ処理システムに対しては、前記シリアルインタフェース手段をデバッグ制御用のデバッグツール若しくはホストシステムに接続可能にするために、データ処理システムの回路基板（50）には、前記データプロセッサに含まれるシリアルインタフェース手段をデバッグツール等に接続するための外部接続コネクタ（51）を設けておけばよい。

【0028】

8

【発明の実施の形態】

《マイクロコンピュータの概要》図1には本発明の一例に係るマイクロコンピュータ1のブロック図が示される。同図に示されるマイクロコンピュータ1は、特に制限されないが、公知の半導体集積回路製造技術によって単結晶シリコンのような1個の半導体基板に形成されている。

【0029】マイクロコンピュータ1は、中央処理装置（CPUとも称する）10、このCPUによってアクセス可能な内部メモリ11、前記CPU10への割込みを制御する割込みコントローラ12、バスコントローラ13、ブレークコントローラ14、システムコントローラ17、及び外部との間でシリアルに情報の入出力を行うシリアルインタフェース回路（SDI）15を備え、それらは代表的に示された内部データバス16D及び内部アドレスバス16Aなどによって接続されている。前記CPU10として示した回路ブロックにディジタル信号処理のための積和演算器などを含めて、マイクロコンピュータ1をディジタル信号処理に特化したディジタル・シグナル・プロセッサの様なデータプロセッサとして構成することも可能である。

【0030】図1に示されるマイクロコンピュータ1は、特に制限されないが、デバッグ支援機能を備えたマイクロコンピュータである。前記、ブレークコントローラ14及びシリアルインタフェース回路15はデバッグを支援するために特別に搭載されている。デバッグ支援機能を備えていないマイクロコンピュータを便宜上、実チップとも称する。実チップと同等の機能を前提にデバッグ支援機能を備えたマイクロコンピュータを便宜上、エバチップ（評価チップ）とも称する。

【0031】エバチップとしてのマイクロコンピュータ1の内部メモリ11は、多くの場合実チップがROM（リード・オンリ・メモリ）で提供する前記CPU10の動作プログラムを格納するための記憶領域をRAM（ランダム・アクセス・メモリ）で提供する。このRAMの領域はユーザメモリ領域（User_MEM）110として図示されている。内部メモリ11において111で示される領域は、デバッグのための制御用プログラム（デバッグプログラム）やその他の制御情報の格納用に割り当てられ、RAMで構成されているデバッグ用RAM領域（ASERAM）とされる。参照符号112で示される領域は、前記CPU10に前記シリアルインタフェース回路15から前記デバッグ用RAM領域111へデバッグプログラムやその他の制御情報を転送させるためのブートプログラムなどを保有するブートプログラム領域（Boot_Program）である。このブートプログラム領域112はROMで構成されている。

【0032】前記バスコントローラ13は、マイクロコンピュータの内外に対するCPU10のバスアクセスに際し、アクセス対象領域に応じたデータアクセス幅やウ

50

(6)

9

エートサイクルの挿入制御などを行う。このバスコントローラ13によるアドレス範囲毎の制御様は、CPU10がレジスタ130に設定した値によって決定される。

【0033】前記ブレークコントローラ14は、後述するブレークモードにおいてCPU10がブレーク条件を設定するレジスタ140を有し、後述するユーザモードにおいて、このレジスタ140に設定されたプログラムアドレスやデータ値などのブレーク条件と一致する状態が内部データバス16Dや内部アドレスバス16Aなどに現れるか否かを監視し、ブレーク条件に一致する状態を検出したとき、ブレーク割り込み要求信号brksを割り込みコントローラ12に与える。

【0034】前記割り込みコントローラ12はCPU10に対する割り込み制御を行う。割り込みコントローラ12には、前記ブレーク割り込み要求信号brks、そして前記シリアルインターフェース回路15から出力される割り込み要求信号SDI_boot、SDI_int、SDI_brkが供給されると共に、図示を省略する外部割り込み要求信号、同じく図示を省略する内部割り込み要求信号が供給される。割り込みコントローラ12は、割り込み要求と割り込み要因情報を対応させる論理回路120を備えると共に、個々の割り込み要求の優先度を決定する論理回路121を有し、割り込み要求があると、その要求に対応される割り込み優先度を参照して、優先度の高い割り込み処理を優先させるように、割り込み要求を受け付け、受け付けた割り込み要求に応ずる割り込み要因情報と割り込み信号をCPU10に与える。CPU10は、その割り込み要因情報からベクタアドレスを演算し、それによって割り込みベクタを取得し、取得した割り込みベクタによって指定される命令アドレスにその処理を分岐させる。このとき、CPU10は、割り込み処理からの復帰に必要な情報の退避を予め行うことは言うまでもない。

【0035】《シリアルインターフェース回路》前記シリアルインターフェース回路15は、マイクロコンピュータ1の外部と非同期でシリアルに情報の入出力を行う回路であり、図2に例示されるように、マイクロコンピュータ1を利用したシステム(ターゲットシステム)5のデバッグに際して、マイクロコンピュータ1をシリアルコントローラ3を介してホストシステム(デバッグツール)4にインターフェースさせるためのデバッグ用回路とされる。図2において2で示されるものは、ターゲットシステム5に実装された外部メモリであり、この外部メモリ2は図1では図示を省略した入出力ポート19を介してバスコントローラ13に接続されている。

【0036】図1に示されるように、前記シリアルインターフェース回路15は、シフト・パラレル変換レジスタ150、セレクタ151、コマンドレジスタSDIR、ステータスレジスタSDSR、データレジスタSDDR、セレクタ152、前記シリアルコントローラ3側と

10

のデータ入出力を制御するスタートコントローラ153、CPU10側とのデータ入出力を制御するレジスタコントローラ157、及びコマンドデコーダ154を有する。特に制限されないが、シフト・パラレル変換レジスタ150及びデータレジスタSDDRは32ビット、コマンドレジスタSDIR及びステータスレジスタSDSRは16ビットとされる。

【0037】前記シフト・パラレル変換レジスタ150は、シリアル入力-パラレル出力(serial-in/parallel-out)とパラレル入力-シリアル出力(parallel-in/serial-out)の機能を備えたシフトレジスタとされ、シリアルデータ入力端子TDIから供給されるシリアルデータをシフトクロック155に同期して順次シフトし、パラレルデータとしてバス156に出力可能にし、また、バス156から供給されるパラレルデータを前記シフトクロック155に同期して順次シフトし、シリアルデータ出力端子TDOからシリアル出力可能にする。

【0038】前記セレクタ151はコマンドレジスタSDIR、ステータスレジスタSDSR及びデータレジスタSDDRの一方のデータ入出力端子を選択的に前記バス156に接続する。前記セレクタ152はそれらレジスタSDIR、SDSR、SDDRの他方のデータ入出力端子を選択的に前記データバス16Dに接続する。

【0039】前記レジスタコントローラ157は、バスコントローラ13を介して供給されるアドレス情報及びリード/ライト信号等を受け、そのアドレス信号をdecodeして前記レジスタSDIR、SDDR、SDSRの選択信号とレジスタに対するリード/ライト信号を生成して前記セレクタ152を制御する。これにより、CPU10は前記レジスタSDIR、SDSR、SDDRを必要に応じてアクセスできる。

【0040】前記コマンドレジスタSDIRは、図3に例示されるように、TS0、TS1、TS2の3ビットに、ブートコマンド、SDI割り込みコマンド又はSDIブレークコマンドが供給される。前記コマンドデコーダ154は、ブートコマンドが供給されると割り込みコントローラ12に割り込み要求信号(SDIブート割り込み要求信号)SDI_bootを与え、同様に、SDI割り込みコマンドが供給されると割り込み要求信号(SDI割り込み要求信号)SDI_intを、また、SDIブレークコマンドが供給されると割り込み要求信号(SDIブレーク割り込み要求信号)SDI_brkを、割り込みコントローラ12に与える。それら割り込み要求に応ずる処理の内容は後で詳述する。

【0041】前記シリアルコントローラ3による転送クロックTCKに同期したシリアルインターフェース回路15へのアクセスと、CPU10によるシリアルインターフェース回路15へのアクセスとは、相互に非同期とされる。このとき、相互間でのデータの受け渡しを制御するために、前記ステータスレジスタSDSRは1ビットの

(7)

II

フラグFLGを有する。このフラグFLGは、データレジスタSDDRに対するCPU10のアクセスとシリアルコントローラ3側からのアクセスとを排他的に制御可能にするための参照情報とされる。すなわち、このフラグFLGは、その論理値0により、CPU10によるデータレジスタSDDRのアクセスが可能であることを示し、シリアルコントローラ3によるデータレジスタSD DRのアクセスが不可能であることを示す。またフラグFLGは、その論理値1により、CPU10によるデータレジスタSDDRのアクセスが不可能であることを示し、前記シリアルコントローラ3によるデータレジスタSDDRのアクセスが可能であることを示す。尚、ステータスレジスタSDSRは、シリアルインターフェース回路15の内部状態を示すその他のビットも備えている。

【0042】シリアルコントローラ3は、前記フラグFLGを観測することによって、シリアルインターフェース回路15へのデータ転送を制御する。フラグFLGが論理値1にされるのを、シリアルデータ出力端子TDOを介して検出すると、シリアルコントローラ3はシリアルデータ入力端子TDIにデータを供給するダウンロードと、シリアルデータ出力端子TDOからデータを出力するアップロードとを実行する。シリアルコントローラ3はフラグFLGが論理値1にされない限りデータ転送を実行しない。すなわち、データレジスタSDDRへのアクセスを行わない。CPU10は前記フラグFLGが論理値1にされるとデータレジスタSDDRへのアクセスが不可能とされ、例えばバスコントローラ13によってデータレジスタSDDRへのアクセスがウェイトされる。シリアルコントローラ3はデータ転送を完了すると、前記フラグFLGを論理値0にクリアする。フラグFLGが論理値0にクリアされると、ウェイトされていたCPU10によるデータレジスタSDDRのアクセスが再開され、CPU10は、データレジスタSDDRに対するリード又はライトアクセスが可能にされる。CPU10によるデータレジスタSDDRのアクセス完了後、CPU10はフラグFLGを論理値1にセットする。この状態をシリアルコントローラ3が検出することにより、シリアルコントローラ3は再びシリアルインターフェース回路15のデータレジスタSDDRをアクセス可能にされる。

【0043】要するに、シリアルコントローラ3はフラグFLGが論理値1のときデータレジスタSDDRをアクセスし、アクセス完了によってフラグFLGを論理値0にクリアする。CPU10はフラグFLGが論理値0にクリアされているときデータレジスタSDDRをアクセスし、アクセス完了によってフラグFLGを論理値1にセットする。このように、フラグFLGを用いた上記規則若しくは制御によって、CPU10とシリアルコントローラ3によって非同期でアクセスされるデータレジスタSDDRのオーバライトを防いで、外部との間で非

12

同期で容易にシリアルデータをやりとりすることができる。

【0044】前記ステートコントローラ153は、それぞれ1ビットのリセット信号/TRST、転送クロック信号TCK、及びステート制御信号TMSを受ける。リセット信号/TRSTはシリアルインターフェース回路15を初期化するリセット信号とされる。転送クロック信号TCKは前記シリアルデータ入出力端子TDI、TDOを介するデータ転送の同期クロック信号とされ、また、ステートコントローラ153の動作基準クロック信号とされる。また、ステートコントローラ153は、シリアルデータ出力端子TDOを参照し、前記フラグFLGがシフト・パラレル変換レジスタ150を介してシリアルデータ出力端子TDOから出力されるとき、出力されたフラグFLGの論理値が1であるか否かを認識できるようになっている。

【0045】前記ステートコントローラ153は、所謂ステートマシンとされ、予め定められた順番でシリアルインターフェース回路15の内部制御状態を遷移させ、このときの遷移方向は前記ステート制御信号TMSの論理値によって決定する。

【0046】図5には、前記ステートコントローラ153による制御状態の遷移図が例示されている。ST1はリセットステート、ST2はアイドルステート、ST3はデータレジスタSDDR(又はステータスレジスタSD SR)の選択ステート、ST4はコマンドレジスタSDIRの選択ステートである。現在のステートから次のステートに進む方向はステート制御信号TMSの論理値1, 0によって決定される。

【0047】ステートST3においてレジスタSDDR又はSDSRの何れを選択するかは、論理値1のフラグFLGがシリアルデータ出力端子TDOを介して外部に出力されたか否かによって決定される。すなわち、ステートコントローラ153は、論理値1のフラグFLGがシリアルデータ出力端子TDOを介して外部に出力されたことを認識することにより、前記ステートST3においてデータレジスタSDDRを選択する。

【0048】前記ステートST3において、TMS=0にされた場合には、レジスタSDDR(SDSR)のデータをシフト・パラレル変換レジスタ150へ出力するキャプチャステートST30、シフト・パラレル変換レジスタ150を制御信号TMS=0のとき転送クロックTCKに同期してシフト動作させるシフトステートST31、作業ステート(動作切り換えのための予備ステート)ST32、ポーズステートST33、作業ステート(動作切り換えのための予備ステート)ST34、及びシフト・パラレル変換レジスタ150からレジスタSD DR(SDSR)へデータを並列出力するアップデートステートST35を、ステート制御信号TMSの論理値50に従って順次採り得るようにされる。これによって、シ

(8)

13

リアルデータ入力端子 TDI に供給されたデータをシフト・パラレル変換レジスタ 150 を介してレジスタ SDDR に取り込み、また、レジスタ SDDR (SDSR) にセットされているデータをシフト・パラレル変換レジスタ 150 を介してシリアルデータ出力端子 TDO から出力することができる。

【0049】例えばシリアルインターフェース回路 15 のリセットによってフラグ FLG は論理値 1 に初期化される。この状態において、ステート ST3 で選択されるレジスタはステータスレジスタ SDSR とされ、キャプチャステート ST30 を介してフラグ FLG が外部に読み出されると、それをステートコントローラ 153 が検出し、今度、ステート ST3 に戻ると、当該ステート ST3 では、データレジスタ SDDR が選択され、シリアルデータ入力端子 TDI から供給されるデータを、シフトステート ST31 やアップデートステート ST35 等を介してデータレジスタ SDDR に格納することができる。特に制限されないが、アップデートステート ST35 を一旦抜けると、その後のステート ST3 におけるレジスタ選択状態はステータスレジスタ SDSR に初期化される。

【0050】コマンドレジスタ SDIR の選択ステート ST4において、TMS=0 にされた場合には、コマンドレジスタ SDIR のコマンドをシフト・パラレル変換レジスタ 150 へ出力するキャプチャステート ST40、シフト・パラレル変換レジスタ 150 のシフトステート ST41、作業ステート（動作切り換えのための予備ステート）ST42、ポーズステート ST43、作業ステート（動作切り換えのための予備ステート）ST44、及びシフト・パラレル変換レジスタ 150 からコマンドレジスタ SDIR へコマンドを出力するアップデートステート ST45 を、ステート制御信号 TMS の論理値に従って順次採り得るようにされる。これによって、シリアルデータ入力端子 TDI に供給されたコマンドをシフト・パラレル変換レジスタ 150 を介してコマンドレジスタ SDIR に取り込むことができる。尚、状態を遷移させる方向によっては、途中の制御状態はダミーにされる場合がある。

【0051】《マイクロコンピュータの動作モードとアドレス空間》図 1において参照符号 17 で示されるものは、システムコントローラである。システムコントローラ 17 は、マイクロコンピュータ 1 の動作モードなどを制御する。この例では、モード信号として、エミュレータモード信号 ASEMD0 が代表的に示されている。特に制限されないが、マイクロコンピュータ 1 のリセット信号/RST もシステムコントローラ 17 に供給されている。

【0052】図 4 に示されるように、ASEMD0=0 のときマイクロコンピュータ 1 は ASE モード（評価モード若しくはデバッグモード）とされ、ASEMD0

14

=1 のときマイクロコンピュータ 1 は本体チップモード（実モード）とされる。

【0053】実モードは、評価チップとしてのマイクロコンピュータ 1 を実チップとして動作させるモード（デバッグモードではない）である。このときの CPU10 のアドレス空間は図 6 に例示されるように、ユーザメモリ領域 (User_MEM) 110 の空間、外部メモリ空間（マイクロコンピュータ 1 の外部メモリ空間）及び周辺モジュール用空間とされ、その他の空間はリザーブ空間（利用不可能な空間）とされる。周辺モジュール用空間は、実チップに含まれる周辺回路例えば前記割り込みコントローラ 12 やバスコントローラ 13 などの内部レジスタ空間などとされる。前記ユーザメモリ領域 110 の空間は、実チップでは内蔵 RAM や内蔵 ROM で構成されるワーク領域、データ一時記憶領域、及びプログラム領域とされる。実モードにおいて CPU10 はユーザメモリ領域 110 の空間又は外部メモリ空間に配置されたプログラム領域から命令をフェッチし、これを実行する。ユーザメモリ領域 110 の先頭には、リセット処理などの例外処理のためのベクタテーブルの領域が割り当てられ、実モードにおいて CPU10 は、リセット（パワーオンリセット、ハードウェアリセット、又はソフトウェアリセット）されると、当該ベクタテーブルの先頭 (H'00000000) に配置されたリセットベクタを取得し、それによって示されるリセット処理プログラムを最初に実行して、プログラム処理を進めていく。

【0054】前記評価モードは、デバッグモードであり、特に、CPU10 が前記実モードと同様にユーザメモリ領域 110 に格納されたユーザプログラムを実行可能な動作モードをユーザモードと称し、前記デバッグ用 RAM 領域 111 に格納されたデバッグプログラムを実行可能な動作モードをブレークモードと称する。

【0055】前記ユーザモードにおける CPU10 のアドレス空間は、図 7 に例示され、実モードに対して何ら変わらない。

【0056】前記ブレークモードにおける CPU10 のアドレス空間は、図 8 に例示され、ユーザモードに対し、エミュレータ空間（内部／外部）が追加されている。前記外部エミュレータ空間は、図示しない ASE 端子がある場合、例えばトレース空間などに利用することができます。前記内部エミュレータ空間は、図 9 に例示される領域が割り当てられる。前記シリアルインターフェース回路 15 の内部レジスタ SDIR, SDSR, SDDR や、ブレークコントローラ 14 の内部レジスタ 140 などは、図 6、図 7、図 8 に示された周辺モジュール用空間にマッピングされる。図 9 のアドレスマップにおいて、R0 退避領域 111A は、ユーザモードからブレークモードへの切り換えに際し、CPU10 の内部レジスタ R0 の値を一時的に退避し、CPU10 がレジスタ R0 を利用して動作モード切り換えに際して付帯する処理

(9)

15

を行っても、ユーザモードにおける切り換え時点のレジスタ R0 の値を後から参照可能にするための領域である。ハードウェアブレーク・ベクタ領域 111B は、ハードウェアブレークが発生したとき、CPU10 のプログラム空間の切り換えに利用されるベクタ格納領域である。ハードウェアブレークは、前記割り込み要求信号 brks, SDI_brk によって発生される。ソフトウェアブレーク・ベクタ領域 111C は、ソフトウェアブレークが発生したとき、CPU10 のプログラム空間の切り換えに利用されるベクタの格納領域である。前記 R0 退避領域 111A、ハードウェアブレーク・ベクタ領域 111B、ソフトウェアブレーク・ベクタ領域 111C はデバッグ用 RAM 領域 111 の一部とされ、RAM で提供されている。SDI ブート・ベクタ領域 112A は、割り込み要求信号 SDI_boot が受け付けられたとき CPU10 が参照するベクタの格納領域である。例えば、この領域 112A に格納されるベクタは、ブートプログラム領域 112 の先頭アドレス H'FFFFFC0 を示し、これによって CPU10 は、ブートプログラム領域 112 に格納されているブートプログラムの実行を開始する (S 6)。

【0057】《SDI_bootによるブレークモード確立のためのブート動作》次に評価モード（デバッグモード）に於けるブレークモードの確立、並びにブレークモードとユーザモードとの間での動作モードの遷移について説明する。

【0058】図 10 には、前記ブレークモードを確立するためのブート動作の制御手順が示され、図 11 にはその時のタイミングチャートが示される。

【0059】デバッグモードを確立するには、前記モード信号 A SEMD0 をローレベルにすると共に、マイクロコンピュータ 1 のリセット信号 /RST とシリアルインターフェース回路 15 のリセット信号 /TRST をそれぞれローレベルにする (S 1)。これにより、シリアルインターフェース回路 15 は初期化され、また、マイクロコンピュータ 1 はリセットホールド状態にされる (S 2)。次いで、リセット信号 /RST をハイレベルにした後、リセット信号 /TRST をハイレベルにすることによってマイクロコンピュータ 1 のリセットホールド状態が維持される (S 3)。この状態で、マイクロコンピュータ 1 はデバッグモードにされ、CPU10 は図 8 に示されるブレークモードのアドレスマッピングを探る。マイクロコンピュータ 1 のリセットホールド状態において、CPU10、割り込みコントローラ 12 及びシリアルインターフェース回路 15 は動作可能にされている。

【0060】この状態で転送ロック TCK に同期して、ステート制御信号 TMS が所要に変化されて、シリアルデータ入力端子 TDI からブートコマンドがコマンドレジスタ SDI_R 供給されると (S 4)、そのブートコマンドがコマンドデコーダ 154 で解読され、それに

(9)

16

よってコマンドデコーダ 154 は、ブート割り込み要求信号 SDI_boot を割り込みコントローラ 12 に出力する (S 5)。

【0061】割り込みコントローラ 12 は、そのブート割り込み要求信号 SDI_boot を受け付けると、当該割り込みに対応される要因情報と割り込み信号を CPU10 に供給する。CPU10 は、その割り込み信号によってリセットホールド状態が解除され、その割り込み要因情報に基づいてベクタアドレスをハードウェア的に取得し、これによって CPU10 は図 9 に示される SDI ブート・ベクタ領域 112A のベクタを取得する。この例に従えば、前記ベクタは、ブートプログラム領域 112 の先頭アドレス H'FFFFFC0 を示し、これによって CPU10 は、ブートプログラム領域 112 に格納されているブートプログラムの実行を開始する (S 6)。

【0062】ブートプログラムは、シリアルデータ入力端子 TDI を介して供給される情報をデバッグ用 RAM 領域 111 に転送制御するプログラムとされる。これにより、シリアルインターフェース回路 15 を介してシリアルデータ入力端子 TDI に供給されるデバッグプログラムやそれに利用するベクタテーブル等の情報が、デバッグ用 RAM 領域 111 にダウンロードされる (S 7)。

【0063】この例に従えば、ブートプログラムの最後には、CPU10 の命令実行状態を、前記デバッグ用 RAM 領域 111 の先頭アドレス H'FFFF800 に分岐させる分岐命令を含んでいる。これにより、ブートプログラムの実行に統一して、CPU10 の命令実行はデバッグ用 RAM 領域 111 のデバッグプログラムに分岐され (S 8)、CPU10 はデバッグプログラムを実行可能な状態に遷移される (S 9)。このようにして、マイクロコンピュータ 1 は評価モードのブレークモードを確立する。

【0064】図 11 において時刻 t1 は、前記リセットホールド状態が達成されるタイミングを示す。リセットホールド状態にされると、時刻 t2 のタイミングでブレークモードにされる。時刻 t3 にリセット信号 /RST をハイレベルにすると、その後、リセット信号 /TRST をハイレベルに変化させることにより、シリアルインターフェース回路 15 のリセット状態が解除される。但し、CPU10 のリセットホールド状態は維持される。時刻 t4 に前記ブートコマンドが発行されると、割り込みによって CPU10 のリセットホールド状態が解除され、CPU10 はブートプログラムを実行し、シリアルデータ入力端子 TDI から入力されたデータをデバッグ用 RAM 領域 111 にダウンロードする。時刻 t5 にブートプログラムの実行を完了すると、前述の通り、CPU10 の命令実行状態は、デバッグ用 RAM 領域 111 にダウンロードされたデバッグプログラムに分岐される。

【0065】デバッグプログラムの内容は、マイクロコンピュータ 1 を適用するシステムの構成に応じて、ユ

(10)

17

ユーザが任意に決定する。

【0066】この例において、デバッグ用RAM領域111の先頭には、前記シリアルインタフェース回路15を介してユーザメモリ領域110のプログラムメモリ空間にユーザプログラム（デバッグ対象プログラム）をダウンロードするための転送制御プログラムを配置することができる。そして、前記デバッグプログラムには、そのような転送制御プログラムを初めとして、マイクロコンピュータ内部の状態を前記シリアルインタフェース回路15を介して外部に出力するためのプログラムや、前記シリアルインタフェース回路15を介して前記ブレーキコントローラ14にブレーキ条件を設定するためのプログラムなどを含めることができる。或いは、そのようなプログラムを選択するための操作を可能にするプログラム等を配置することもできる。また、CPU10の命令実行状態を、前記ユーザメモリ領域111や外部メモリの所望のプログラム格納領域のアドレスに分岐させるための操作を行う命令などを配置すれば、ブレーキモードからユーザモードに移行することも可能である。

【0067】《SDI_brkによるブレーキ》図12には割り込み要求信号SDI_brkによるブレーキモードへの遷移制御手順が示され、図13にはそのときのタイミングチャートが示される。

【0068】ユーザモードでは、ブレーキコントローラ14及びシリアルインタフェース回路15をアクセスすることはできない。ユーザモードにおいては、ブレーキモードでブレーキコントローラ14に設定されたブレーキ条件が成立すると、マイクロコンピュータ1は、ブレーキ割り込み要求信号brksによってブレーキモードに移行される。ユーザモードにおいて、SDI_brkによるブレーキ割り込み要求、そして後述するSDI_intによる割り込み要求は受付可能にされている。実モードではSDI_brkによるブレーキ割り込み要求は受け付けられない。

【0069】SDIブレーキ割り込み要求信号SDI_brkによるブレーキを起動するには先ず、リセット信号/TRSTをローレベルとし、シリアルインタフェース回路15を初期化し（S10）、その後で、リセット信号/TRSTをハイレベルに戻す（S11）。そして、ブレーキコマンドをシリアルデータ入力端子TDIから供給すると（S12）、そのコマンドがコマンドデコーダ154でデコードされ、その結果、割り込み要求信号SDI_brkが割り込みコントローラ12に出力される（S13）。

【0070】割り込みコントローラ12は、そのブレーキ割り込み要求を受け付けると、当該割り込みに対応される要因情報と割り込み信号をCPU10に供給する。CPU10は、その割り込み要因情報に基づいてベクタアドレスをハードウェア的に取得し、これによってCPU10は図9に示されるハードウェアブレーキ・ベクタ領域111Bのベクタを取得する。このハードウェアブレーキ・ベクタ領域111Bはデバッグ用RAM領域1

18

11の一部の領域が割り当てられ、それに設定されているベクタに応じて、CPU10の処理がデバッグプログラムの所定の処理に分岐され（S14）、CPU10は分岐先のプログラムを実行してエミュレーション動作を行う（S15）。例えば、シリアルインタフェース回路15を介してブレーキコントローラ14にブレーキ条件を設定したり、CPU10の内部レジスタの値をシリアルインタフェース回路15を介して外部に読み出したりする動作を行う。

【0071】図13において、マイクロコンピュータ1がユーザモードのとき（ブレーキモードの場合を除く）、リセット信号/TRSTにより時刻t1にシリアルインタフェース回路15がリセットされている。マイクロコンピュータ1のその他の回路は、それによってリセットされない。そしてシリアルインタフェース回路15のリセットを解除した後、時刻t2に、シリアルデータ入力端子TDIからシリアルインタフェース回路15にブレーキコマンドが供給される。これによって上述の通り、ブレーキ割り込み要求信号SDI_brkによって、マイクロコンピュータ1はブレーキモードに遷移し、CPU10はデバッグ用RAM領域111のデバッグプログラムの所定の処理に分岐し、例えば、シリアルインタフェース回路15を介して、新たなブレーキ条件設定の為のデータを受け取ったりする。

【0072】《SDI_intによる割り込み》図14には割り込み要求信号SDI_intによる割り込み制御手順が示され、図15にはそのときのタイミングチャートが示される。

【0073】割り込み要求信号SDI_intによる割り込みは、実モード又はユーザモードにて受付可能にされる（ブレーキモードでは受付不可能）。この割り込みによって参照されるベクタテーブルはユーザメモリ領域110に形成されなければならないからである。

【0074】SDI割り込み要求信号SDI_intによる割り込みを起動するには先ず、リセット信号/TRSTをローレベルとし、シリアルインタフェース回路15を初期化し（S20）、その後で、リセット信号/TRSTをハイレベルに戻す（S21）。そして、割り込みコマンドをシリアルデータ入力端子TDIから供給すると（S22）、そのコマンドがコマンドデコーダ154でデコードされ、その結果、割り込み要求信号SDI_intが割り込みコントローラ12に出力される（S23）。

【0075】割り込みコントローラ12は、その割り込み要求を受け付けると、当該割り込みに対応される要因情報と割り込み信号とをCPU10に供給する。CPU10は、その割り込み要因情報に基づいてベクタアドレスを取得し、これによってCPU10はユーザメモリ領域110からベクタを取得する。そのベクタは、ユーザプログラムに応じて任意に決定されており、そのベクタに応じて、CPU10の処理は、ユーザプログラムの所

(11)

19

定の処理に分岐され(S 2 4)、CPU 1 0は分岐先のユーザプログラムを実行する(S 2 5)。

【0076】図15において、マイクロコンピュータ1がユーザモードのとき(本体チップモードの場合を除外しない)、リセット信号/TRSTにより時刻t1にシリアルインターフェース回路15がリセットされている。マイクロコンピュータ1のその他の回路は、それによってリセットされない。そしてシリアルインターフェース回路15のリセットを解除した後、時刻t2に、シリアルデータ入力端子TDIからシリアルインターフェース回路15に割り込みコマンドが供給される。これによって上述の通り、割り込み要求信号SDI_intによって、CPU 1 0の処理は割り込みが要求された処理に分岐される。

【0077】図16には前記マイクロコンピュータ1を適用したターゲットシステムをデバッグする場合の全体的なシステム構成例が示される。この図に示される構成は図2に対応されるものである。ターゲットシステム5とデバッグツール4とのインターフェースはシリアルコントローラ3と前記シリアルインターフェース回路15とを介して行われる。デバッグツール4は例えばエミュレーションプログラムを実行することができる携帯用のパソコンコンピュータを一例としている。シリアルコントローラ3はデバッグツール4からの情報転送の指示に従つて、前記信号/TRST, TCK, TMSを制御すると共に、シリアルデータ入力端子TDIにデータをシリアル供給し、シリアルデータ出力端子TDOから出力されるシリアルデータを取り込み制御する。シリアル転送のプロトコルは、上記の例では回路基板に実装された多数の半導体集積回路のテストに利用される所謂JTAGに準拠しているが、それに限定されるものではない。

【0078】図16において、ターゲットシステム5の回路基板5 5には、前記マイクロコンピュータ1のシリアルインターフェース回路15をシリアルコントローラ3と接続するためのが外部接続コネクタ5 1が配置されている。この外部接続コネクタ5 1は、5本の信号(/TRST, TCK, TMS, TDI, TDO)に対応されるだけではなく、回路基板上で大きな面積を必要としない。携帯通信端末や携帯機器のような小型のターゲットシステムに好適である。この外部接続コネクタ5 1はインターフェイスケーブル3 0を介してシリアルコントローラ3に接続される。

【0079】例えば、ターゲットシステム5が携帯電話機のような携帯通信端末である場合に、そのシステムデバッグ若しくはソフトウェアデバッグを、実使用環境に極力近付けて屋外で行う様な場合にも、ターゲットシステム5とデバッグツールとのインターフェースに必要な信号線の本数が極力少なくされているから、そのような利用形態にも即座に対応することができるという、機動性に富む。

【0080】以上説明したように、ターゲットシステム

20

5をソフトウェアデバッグ若しくはシステムデバッグする場合、デバッグ対象とされるユーザプログラムをマイクロコンピュータ1に実行させ、その過程で得られた情報を外部で参照しながら、当該ユーザプログラムの実行軌跡をたどって、デバッグを行う。前記ブレークコントローラ1 4の内部に、前記ユーザプログラム実行軌跡をたどるためのトレース情報を保有するものもある。そのようなユーザプログラムの実行過程で現れる情報を外部で参照したり、あるいは、ユーザプログラムの動作条件等を設定するには、ユーザプログラムとは異なるデバッグプログラムをマイクロコンピュータ1に実行せざることが必要になる。前記ブートプログラムは、そのようなデバッグプログラムを、ユーザプログラムの格納領域1 1 0とは異なるデバッグ用RAM領域1 1 1にダウンロードするためのCPU 1 0の動作プログラムである。ターゲットシステム5の構成に応じて必要とされるデバッグプログラムを予め所要に作成しておき、前記SDIブートコマンドをデバッグツール4からシリアルインターフェース回路15に発行することにより、マイクロコンピュータ1はブートプログラムを起動し、シリアルインターフェース回路15に供給されるデバッグプログラムを、デバッグ用RAM領域1 1 1にダウンロードする。したがって、デバッグに際して、ユーザプログラムの実行過程で現れる情報を外部で参照したり、あるいは、ユーザプログラムの動作条件等を設定するとき、その動作の内容は、ダウンロードされたデバッグプログラムの内容に従つて行われることになる。これにより、そのプログラムの内容次第でどのようなデバッグ動作も行うことができ、デバッグ動作の可変性を保証する事ができる。マイクロコンピュータ1によるデバッグ動作が、シリアルインターフェース回路15で受け付け可能なコマンドの種類によって制限されることはない。

【0081】デバッグツール4とのインターフェースにはシリアルインターフェース回路15を利用し、デバッグの為のインターフェース用端子の数を少なくしており、そのような制限の下においても、デバッグの可変性と、良好な使い勝手とを両立することができる。

【0082】携帯通信端末若しくは携帯機器のようなターゲットシステム5に対し、システムデバッグ若しくはソフトウェアデバッグを、実使用環境に極力近付けて屋外で行う様な場合にも、デバッグの為のインターフェース用端子の数が少なく、また、デバッグの可変性や良好な使い勝手も保証されているので、即座に対応することができるという、機動性に富んでいる。

【0083】前記ブートプログラムには、その最後に、CPU 1 0の命令実行を、前記デバッグ用RAM領域1 1 1に転送された前記デバッグプログラムの所定のアドレス、例えばH'FFFF800に分岐させる命令を含めることができる。これにより、ダウンロードされたデバッグプログラムを自動的に起動して、初期的な制御動作を即

(12)

21

座に開始することができる。

【0084】初期的な動作としてデバッグ対象とされるユーザプログラムのダウンロードを想定すると、前記内部メモリ11は書き換え可能なユーザメモリ領域110を有し、前記デバッグプログラムは、前記ユーザメモリ領域110へユーザプログラムなどを転送させる転送制御プログラムを含むことができる。

【0085】デバッグに際してユーザプログラムをマイクロコンピュータ1内部のユーザメモリ領域110に配置することができるので、実際と同じ動作速度でデバッグの為にユーザプログラムを実行することができる。

【0086】前記シリアルインターフェース回路15は、外部から供給されたSDIブレークコマンドに基づいて、前記割込みコントローラ12にSDI割込み要求信号(SDI_brk)を出力し、前記割込みコントローラ12は、その割込み要求信号に応答して、CPU10に、前記デバッグプログラムの所定の命令を実行させる割込み制御情報をCPU10に与える。したがって、ユーザプログラムの実行途上等で、CPU10の動作プログラムを前記デバッグプログラムに切り換える動作(ブレーク動作)をホスト装置4からシリアルインターフェース回路15を介して任意に行うことができる。

【0087】このとき、前記デバッグ用RAM領域111には、SDIブレークコマンドに基づく割り込み制御情報を受け取ったCPU10が前記デバッグプログラムの所定の命令アドレスを取得するために利用するハードウェアブレーク・ベクタ領域111Bを割り当てる。この領域は前記デバッグプログラムに従ってCPU10が書き換え可能な領域であるから、そのようなブレークコマンドの発行によって実行させたいデバッグ処理も任意に決定することができる。

【0088】前記シリアルインターフェース回路15は、外部から供給されたSDI割り込みコマンドに基づいて、前記割込みコントローラ12にSDI割込み要求信号SDI_intを出力し、前記割込みコントローラ12は、その割込み要求信号に応答して、CPU10に、ユーザメモリ領域1110又はその他のユーザメモリ空間のプログラムを実行させる割込み制御情報をCPU10に与える。これにより、ユーザプログラムの実行途上等で、ホスト装置4からシリアルインターフェース回路15を介して任意に割り込みを行うことができる。

【0089】シリアルインターフェース回路15を介して、マイクロコンピュータ1の外部と非同期でシリアルに情報の入出力をを行うとき、フラグFLGの状態に応じてCPU10によるデータレジスタSDDRのアクセスを許容することにより、マイクロコンピュータ1の外部とCPU10によって非同期でアクセスされるデータレジスタSDDRのオーバライドを防いで、外部との間で、非同期で容易にデータをシリアルにやりとりすることができます。

22

【0090】ステートコントローラ153は、論理値1にされた前記フラグFLGがマイクロコンピュータ1の外部に出力された状態を検出して、そのとき前記データレジスタSDDRをマイクロコンピュータ1の外部からアクセス可能に選択すると共に、外部によるデータレジスタSDDRのアクセス完了に伴って前記フラグFLGを論理値0に変更可能にするから、マイクロコンピュータ1の外部へフラグFLGの情報を出力するのに専用信号線を用いず、シリアルデータ入出力のための信号線を用いることができる。

【0091】以上本発明者によってなされた発明を実施形態に基づいて具体的に説明したが、本発明はそれに限定されるものではなく、その要旨を逸脱しない範囲において種々変更可能であることは言うまでもない。

【0092】例えば、データプロセッサはマイクロコンピュータに限定されず、ディジタル信号処理演算に特化したディジタル・シグナル・プロセッサ、マイクロプロセッサ、機器制御用に特化したマイクロコントローラ等の論理LSIに広く適用することができる。それらデータプロセッサに内蔵される回路モジュールは上記実施例に限定されない。

【0093】

【発明の効果】本願において開示される発明のうち代表的なものによって得られる効果を簡単に説明すれば下記の通りである。

【0094】すなわち、ブートプログラムによってダウンロードされるデバッグ制御用プログラムの内容次第でどのようなデバッグ動作も行うことができ、デバッグ動作の可変性を保証する事ができる。データプロセッサによるデバッグ動作が、シリアルインターフェース手段で受け付け可能なコマンドの種類によって制限されることはない。

【0095】ホストシステムとのインターフェースにはシリアルインターフェース手段を利用し、デバッグの為のインターフェース用端子の数を少なくしており、そのような制限の下においても、デバッグの可変性と、良好な使い勝手とを両立することができる。

【0096】携帯通信端末若しくは携帯機器のようなターゲットシステムに対し、システムデバッグ若しくはソフトウェアデバッグを、実使用環境に極力近付けて屋外で行う様な場合にも、デバッグの為のインターフェース用端子の数が少なく、また、デバッグの可変性や良好な使い勝手も保証されているので、即座に対応することができるという、機動性を得ることができる。

【0097】また、前記ブートプログラムの最後に、中央処理装置の命令実行を、デバッグ制御用プログラムの所定のアドレスに分岐させる命令を含めることにより、ダウンロードされたデバッグ制御用プログラムを自動的に起動して、初期的な制御動作を即座に開始することができる。

(13)

23

【0098】デバッグに際してユーザプログラムをデータプロセッサ内部のメモリに配置することにより、実際と同じ動作速度でデバッグの為にユーザプログラムを実行することができる。

【0099】外部からシリアルインタフェース手段に供給される第2のコマンド(SDIブレークコマンド)に応答する割り込み要求によって中央処理装置に、デバッグ制御用プログラムの所定の命令を実行させることにより、ユーザプログラムの実行途上等で、中央処理装置の動作プログラムを前記デバッグ用制御プログラムに切り換える動作(ブレーク動作)をホスト装置側からシリアルインタフェース手段を介して任意に行うことができる。

【0100】外部からシリアルインタフェース手段に供給される第3のコマンド(SDI割り込みコマンド)に応答する割り込み要求によって中央処理装置を、ユーザメモリ空間の所定のプログラムに分岐可能にすることにより、ユーザプログラムの実行途上等で、ホスト装置側からシリアルインタフェース手段を介して中央処理装置に任意に割り込みを発生させることができる。

【0101】シリアルインタフェース手段を介して、データプロセッサの外部と非同期でシリアルに情報の入出力を行うとき、フラグ(FLG)の状態に応じて中央処理装置によるデータレジスタのアクセスを許容することにより、データプロセッサの外部と中央処理装置によって非同期でアクセスされるデータレジスタ(SDDR)のオーバライドを防いで、外部との間で、非同期で容易にデータをシリアルにやりとりすることができる。

【図面の簡単な説明】

【図1】本発明の一例に係るマイクロコンピュータのブロック図である。

【図2】図1のマイクロコンピュータを適用したターゲットシステムをデバッグツールにインターフェースした状態の一例を示すブロック図である。

【図3】シリアルインタフェース回路に含まれるコマンドレジスタの一例を示す説明図である。

【図4】エミュレータモード信号ASEMODによって設定されるマイクロコンピュータの動作モードの一例を示す説明図である。

【図5】シリアルインタフェース回路に含まれるステートコントローラによる状態遷移制御の遷移図である。

【図6】実モードにおけるCPUのアドレス空間を示す説明図である。

【図7】ユーザモードにおけるCPUのアドレス空間を示す説明図である。

【図8】ブレークモードにおけるCPUのアドレス空間を示す説明図である。

【図9】ブレークモードにおいて割り当てられる内部エミュレータ空間の詳細を示す説明図である。

【図10】ブレークモードを確立するための割り込み要

24

求信号SDI_bootによるブート動作の制御手順を示す一例フローチャートである。

【図11】ブレークモードを確立するための割り込み要求信号SDI_bootによるブート動作の一例を示すタイミングチャートである。

【図12】割り込み要求信号SDI_brkによるブレークモードへの遷移の制御手順を示す一例フローチャートである。

【図13】割り込み要求信号SDI_brkによるブレークモードへの遷移動作の一例を示すタイミングチャートである。

【図14】割り込み要求信号SDI_intによる割り込みの制御手順を示す一例フローチャートである。

【図15】割り込み要求信号SDI_intによる割り込みの制御動作の一例を示すタイミングチャートである。

【図16】図1のマイクロコンピュータを適用したターゲットシステムをデバッグツールにインターフェースした状態の一例を示す斜視図である。

【符号の説明】

- 1 マイクロコンピュータ
- 2 外部メモリ
- 3 シリアルコントローラ
- 4 デバッグツール
- 5 ターゲットシステム
- 50 回路基板
- 51 外部接続コネクタ
- 10 CPU(中央処理装置)
- 110 ユーザメモリ領域
 - 111 デバッグ用RAM領域
 - 111B ハードウェアブレーク・ベクタ領域
 - 112 ブートプログラム領域
 - 112A SDIブート・ベクタ領域
- 11 内部メモリ
- 12 割り込みコントローラ
- 13 パスコントローラ
- 14 ブレークコントローラ
- 15 シリアルインタフェース回路
- /TRST リセット信号
- TCK 転送クロック信号
- TMS ステート制御信号
- TDI シリアルデータ入力端子
- TDO シリアルデータ出力端子
- 150 シフト・パラレル変換レジスタ
- 153 ステートコントローラ
- 154 コマンドデコーダ
- SDIR コマンドレジスタ
- SDSR 素ステータスレジスタ
- FLG フラグ
- SDDR データレジスタ
- 50 SDI_boot SDIブート割り込み要求信号

(14)

25

SDI_int SDI割り込み要求信号

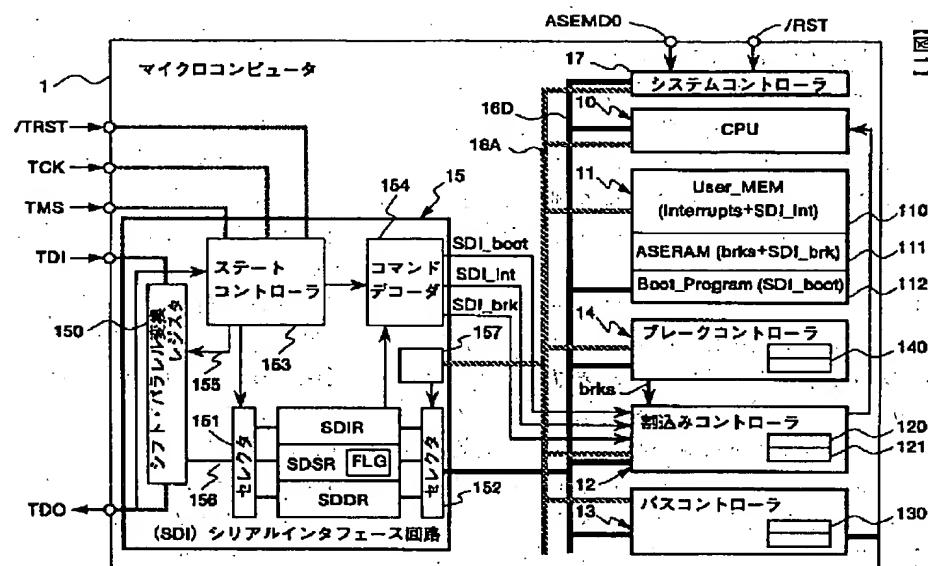
SDI_brk SDIブレーク割り込み要求信号

16D 内部データバス

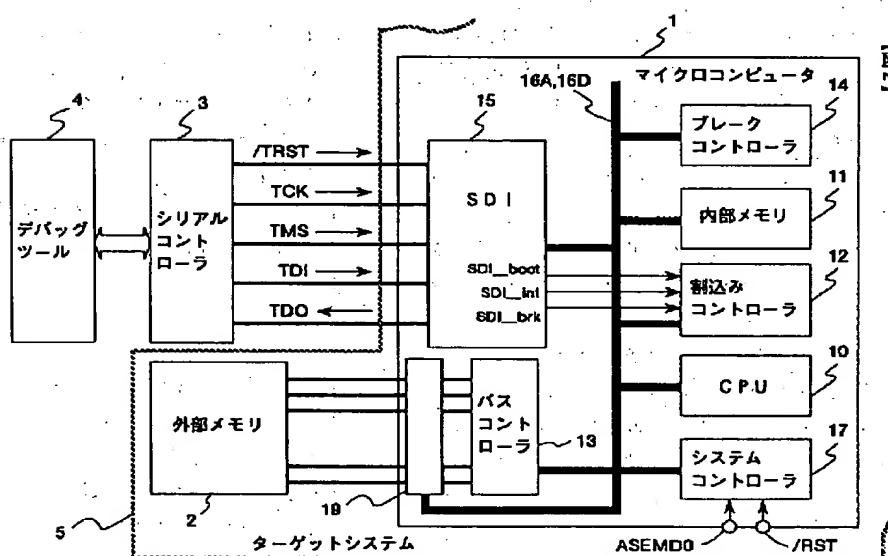
16A 内部アドレスバス

26

【図1】



【図2】



【図4】

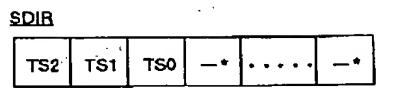
【図4】

ASEMD0	Comment
0	ASEモード (評価チップモード)
1	本体チップモード (実チップモード)

(15)

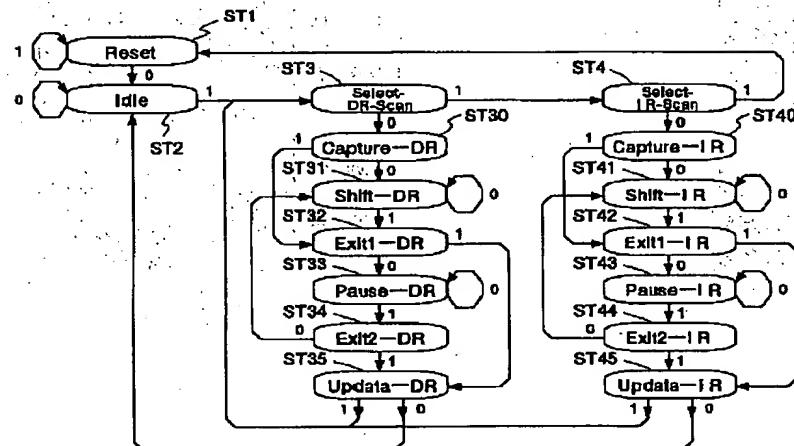
【図3】

【図3】



TS2	TS1	TS0	Command
1	0	0	ブートコマンド
1	0	1	SDI割込コマンド
1	1	0	SDIブレークコマンド

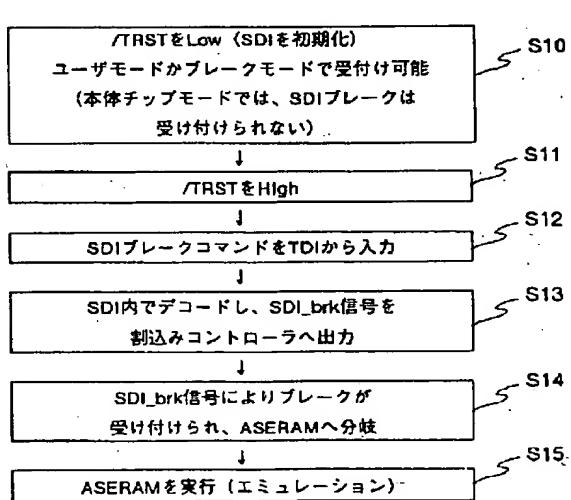
【図5】



【図12】

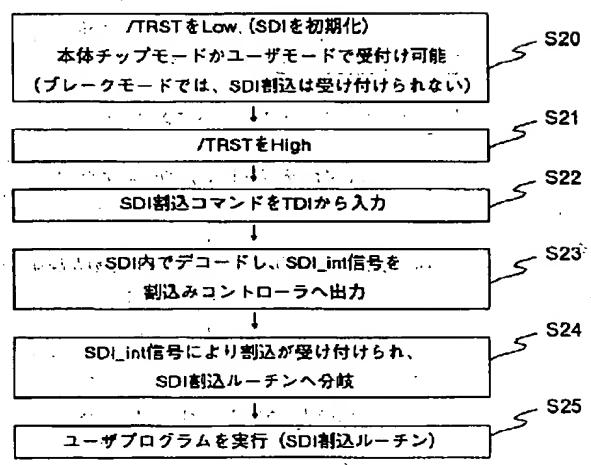
【図12】

SDIブレークフロー



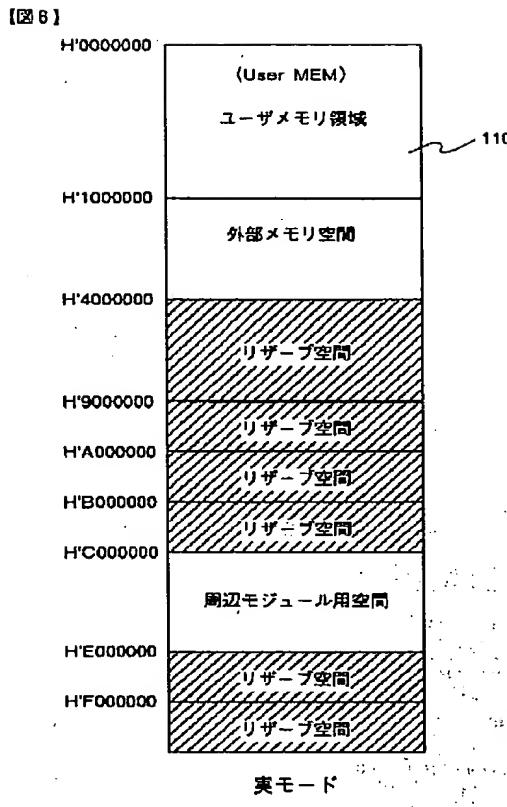
【図14】

SDI割込フロー

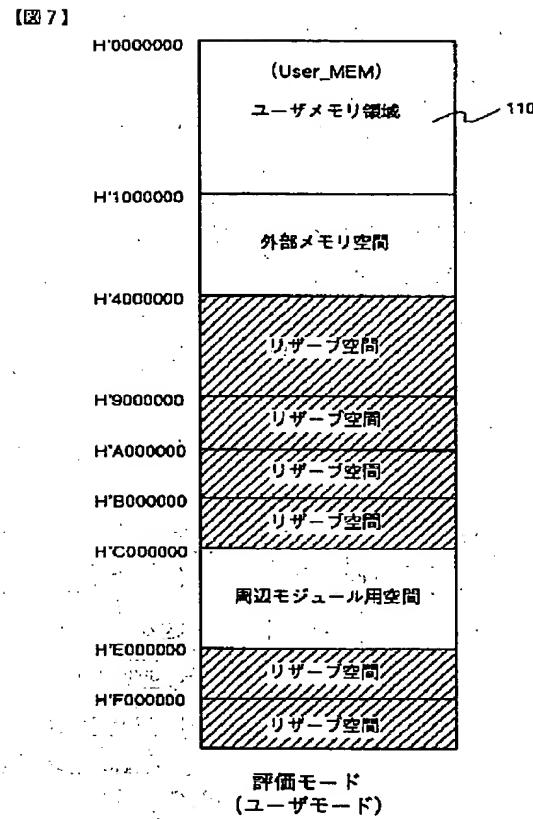


(16)

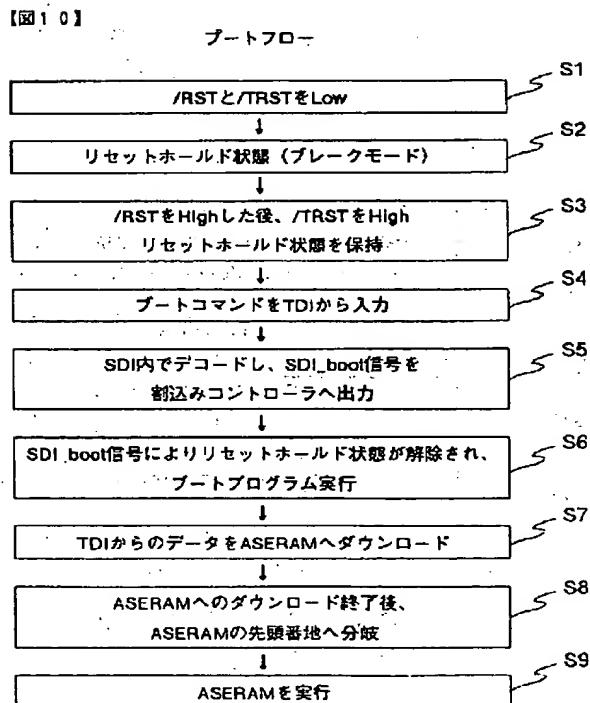
【図 6】



【図 7】

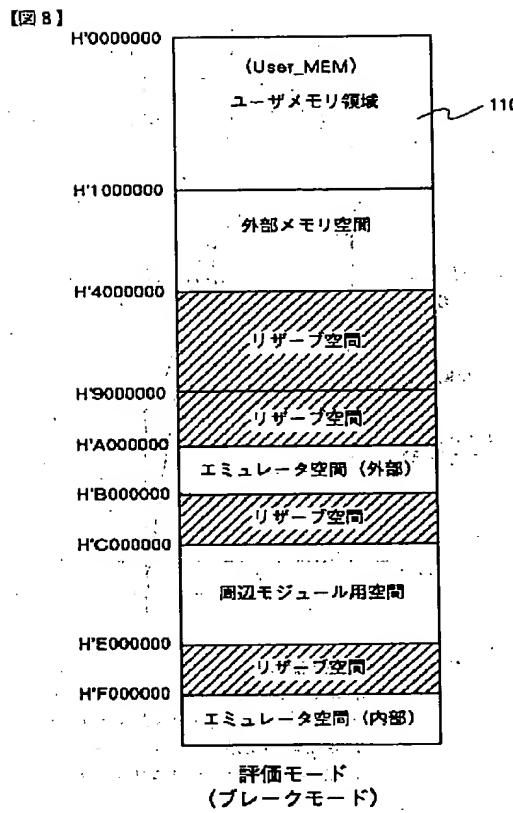


【図 10】

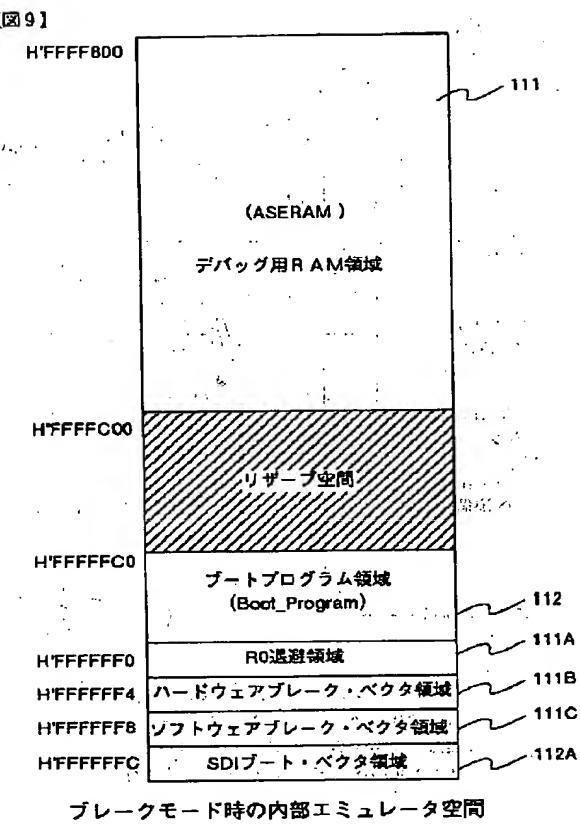


(17)

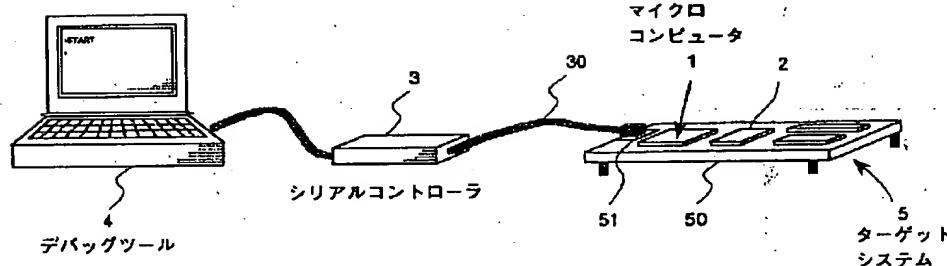
【図8】



【図9】



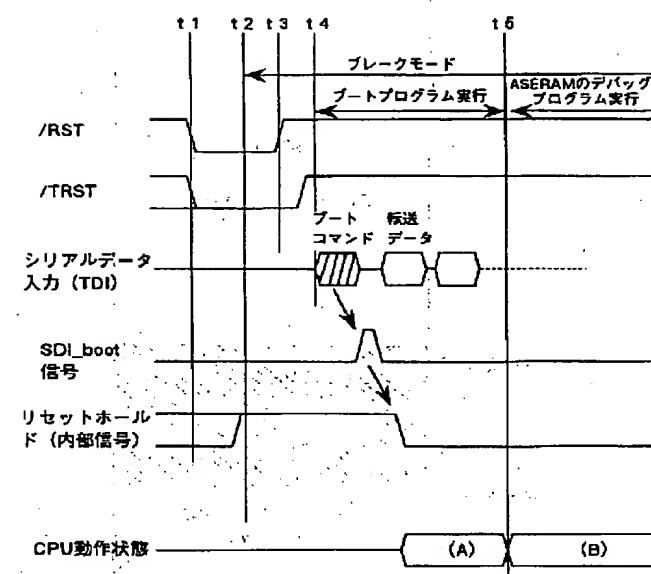
【図16】



(18)

【図11】

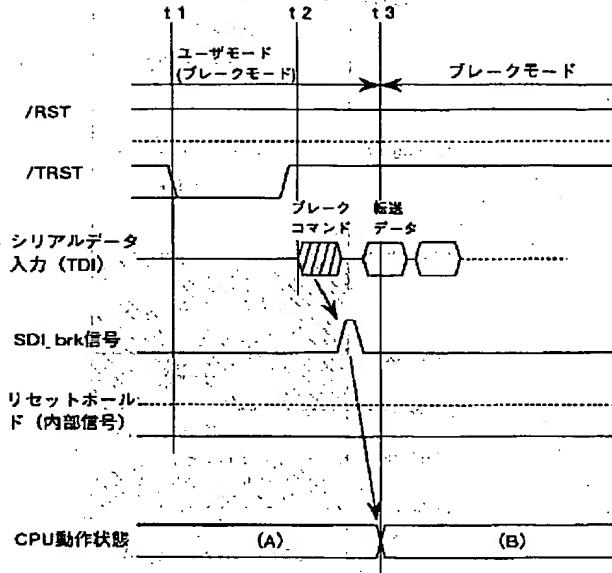
【図1.1】



- (A) ブートプログラム実行 (Boot_Program)
1.ASERAMへブート (ブレークモード)
2.ブート完了後、ASERAMへ分岐
(B) ASERAMのデバッグプログラム実行

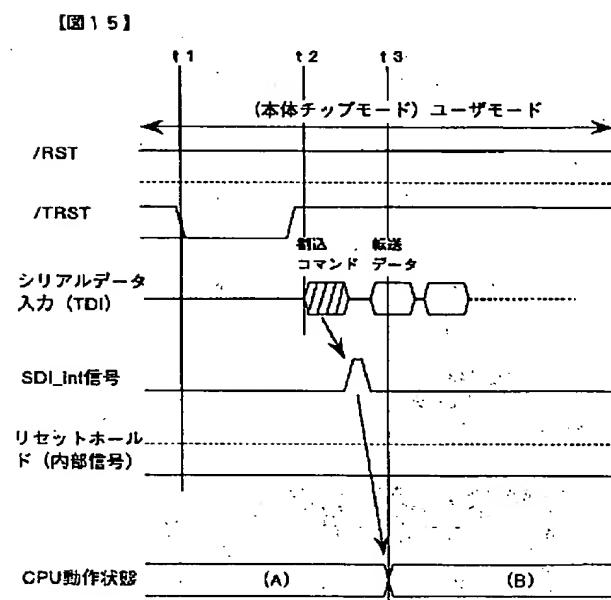
【図13】

【図1.3】



- (A) 通常動作状態
(B) ASERAM実行
1.ハードウェアブレークベクタをリード
2.ASERAMへ分岐
3.ASERAMのデバッグプログラム実行

【図15】



- (A) 通常動作状態
(B) SDI割込
1.SDI割込みベクタリード
2.SDI割込みベクトルーチン実行

【公報種別】特許法第17条の2の規定による補正の掲載
【部門区分】第6部門第3区分
【発行日】平成15年6月13日(2003.6.13)

【公開番号】特開平9-319727
【公開日】平成9年12月12日(1997.12.12)
【年通号数】公開特許公報9-3198
【出願番号】特願平8-138111

【国際特許分類第7版】

G06F 15/78 510
11/22 340
11/28
13/38 350

【F I】

G06F 15/78 510 K
11/22 340 A
11/28 L
13/38 350

【手続補正書】

【提出日】平成15年2月26日(2003.2.26)

【手続補正1】

【補正対象書類名】明細書

【補正対象項目名】特許請求の範囲

【補正方法】変更

【補正内容】

【特許請求の範囲】

【請求項1】 中央処理装置、この中央処理装置によってアクセス可能な記憶手段、前記中央処理装置への割込みを制御する割込みコントローラ、外部との間でシリアルに情報の入出力を行うシリアルインタフェース手段、及びそれらを接続するための内部バスを含んで半導体集積回路化されて成るデータプロセッサであって、前記記憶手段は、書き換え可能な第1の記憶領域を有し、前記中央処理装置は、前記シリアルインタフェース手段が受けた第1のコマンドの解読結果に基づいて前記シリアルインタフェース手段から前記第1の記憶領域へデバッグ制御プログラムを転送し、前記シリアルインタフェース手段が受けた第2のコマンドの解読結果に基づいて前記第1の記憶領域にあるデバッグ制御プログラムの所定命令を実行することを特徴とするデータプロセッサ。

【請求項2】 請求項1において、

前記シリアルインタフェース手段は、前記第1のコマンドの解読結果に基づいて前記割込みコントローラに第1の割込み要求信号を出力し、前記第2のコマンドに基づいて前記割込みコントローラに第2の割込み要求信号を出力し、

前記割込みコントローラは、前記第1の割込み要求信号

に応答して、中央処理装置に、第1の割込み制御情報を与え、前記第2の割込み要求信号に応答して、中央処理装置に、第2の割込み制御情報を与えるものであることを特徴とするデータプロセッサ。

【請求項3】 請求項2において、

前記記憶領域は、前記中央処理装置に前記シリアルインタフェース手段から前記第1の記憶領域へ前記デバッグ制御プログラムを転送させるブートプログラムを保有する第2の記憶領域とを含み、

前記中央処理装置は、前記第1の割り込み制御情報に基づいて前記ブートプログラムを実行することを特徴とするデータプロセッサ。

【請求項4】 請求項3において、

前記第2の記憶領域は更に、前記第1の割り込み制御情報を受け取った中央処理装置が前記ブートプログラムの先頭の命令アドレスを取得するために利用する第1のベクタを含んで成るものであることを特徴とするデータプロセッサ。

【請求項5】 請求項4において、

前記ブートプログラムは、その後に、中央処理装置の命令実行を、前記第1の記憶領域に転送された前記デバッグ制御用プログラムの所定のアドレスに分岐させる命令を含んで成るものであることを特徴とするデータプロセッサ。

【請求項6】 請求項5において、

前記記憶手段は書き換え可能な第3の記憶領域を有し、前記デバッグ制御用プログラムは、前記シリアルインタフェース手段から前記第3の記憶領域へデータを転送させる転送制御プログラムを含んで成るものであることを特徴とするデータプロセッサ。

(2)

【請求項 7】 請求項 6において、

前記第 1 の記憶領域は更に、前記第 2 の割り込み制御情報を受け取った中央処理装置が前記デバッグ制御用プログラムの所定の命令アドレスを取得するために利用する第 2 のベクタを含んで成るものであることを特徴とするデータプロセッサ。

【請求項 8】 中央処理装置、この中央処理装置によってアクセス可能な記憶手段、及び前記中央処理装置への割込みを制御する割込みコントローラを含んで半導体集積回路化されて成るデータプロセッサにおいて、

データプロセッサの外部と前記中央処理装置との双方から夫々操作可能にされたフラグ、データプロセッサの外部と前記中央処理装置の双方から夫々アクセス可能にされたデータレジスタ、及び外部から供給されるコマンドの解読結果に応じた割込み要求信号を前記割込みコントローラに与えるコマンドデコーダを含み、データプロセッサの外部と非同期でシリアルに情報の入出力を行うシリアルインタフェース手段を設け、

前記記憶手段は、前記中央処理装置のよって前記シリアルインタフェース手段からデバック制御プログラムが転送される書き換え可能な第 1 の記憶領域とを含み、外部から供給されるコマンドに含まれる第 1 のコマンドに従って、前記中央処理装置は、前記フラグが第 1 の状態であることを条件に前記データレジスタをアクセスし、前記データレジスタに対するアクセス完了に伴って前記フラグを第 2 の状態に変更するものであることを特徴とするデータプロセッサ。

【請求項 9】 請求項 8において、

第 2 の状態にされた前記フラグがデータプロセッサの外部に出力された状態を検出して、前記データレジスタをデータプロセッサの外部からアクセス可能に制御すると共に、データプロセッサによるデータレジスタのアクセス完了に伴って前記フラグを第 1 の状態に変更可能にするステードコンドローラを更に備えて成るものであることを特徴とするデータプロセッサ。

【請求項 10】 制御信号端子とデータ入出力端子とを有するシリアルインタフェース手段と、

2

前記シリアルインタフェース手段に接続され、第 1 領域と第 2 領域を有する記憶手段と、

中央処理装置とを具備し、

前記シリアルインタフェース手段が前記制御信号端子から制御要求信号を受けた場合に、前記シリアルインタフェース手段は、前記データ入力端子から前記中央処理装置がデバックを実行するための第 1 プログラムを受け取り、

前記記憶手段は、前記第 2 領域へ前記第 1 プログラムを記憶し、前記第 1 プログラムに従ってデバック機能を実行させるための第 2 プログラムを前記第 1 領域に記憶し、

前記中央処理装置は、前記シリアルインタフェース手段から第 2 信号を受けたのに対応して前記第 1 領域の第 1 の所定アドレスへ分岐することを特徴とするデータプロセッサ。

【請求項 11】 請求項 10において、

前記データプロセッサは、前記シリアルインタフェース手段及び前記中央処理装置とに接続される割込み制御回路を更に有し、

前記中央処理装置が前記第 1 領域の第 2 の所定アドレスを実行した場合又は前記シリアルインタフェース手段が第 3 制御要求信号を受けた場合に、前記割り込み制御回路は、前記中央処理装置に割り込み要求信号を送信し、前記中央処理装置は、前記第 2 領域の第 3 の所定アドレスに分岐し、前記第 1 プログラムを実行し、前記シリアルインタフェース手段に所定のデータを送信し、

前記シリアルインタフェース手段は、前記データ入出力端子に前記所定のデータを出力することを特徴とするデータプロセッサ。

【請求項 12】 請求項 10 又は 11において、

前記シリアルインタフェース手段の前記データ入出力端子に前記第 1 プログラムを受けた後に、前記データ入出力端子に第 1 データを受けた場合は、前記第 1 データは、前記シリアルインタフェース手段から前記第 1 領域に送られることを特徴とするデータプロセッサ。

**This Page is Inserted by IFW Indexing and Scanning
Operations and is not part of the Official Record**

BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images include but are not limited to the items checked:

- BLACK BORDERS**
- IMAGE CUT OFF AT TOP, BOTTOM OR SIDES**
- FADED TEXT OR DRAWING**
- BLURRED OR ILLEGIBLE TEXT OR DRAWING**
- SKEWED/SLANTED IMAGES**
- COLOR OR BLACK AND WHITE PHOTOGRAPHS**
- GRAY SCALE DOCUMENTS**
- LINES OR MARKS ON ORIGINAL DOCUMENT**
- REFERENCE(S) OR EXHIBIT(S) SUBMITTED ARE POOR QUALITY**
- OTHER: _____**

IMAGES ARE BEST AVAILABLE COPY.

As rescanning these documents will not correct the image problems checked, please do not report these problems to the IFW Image Problem Mailbox.